Biol. Cybern. 87, 116–140 (2002)
DOI 10.1007/s00422-002-0318-1

Biological Cybernetics

Schema design and implementation of the grasp-related mirror neuron system

Erhan Oztop, Michael A. Arbib

USC Brain Project, University of Southern California, Los Angeles, CA 90089-2520, USA

Received: 6 August 2001 / Accepted in revised form: 5 February 2002

Abstract. Mirror neurons within a monkey's premotor area F5 fire not only when the monkey performs a certain class of actions but also when the monkey observes another monkey (or the experimenter) perform a similar action. It has thus been argued that these neurons are crucial for understanding of actions by others. We offer the hand-state hypothesis as a new explanation of the evolution of this capability: the basic functionality of the F5 mirror system is to elaborate the appropriate feedback – what we call the *hand state* – for opposition-space based control of manual grasping of an object. Given this functionality, the social role of the F5 mirror system in understanding the actions of others may be seen as an exaptation gained by generalizing from one's own hand to an other's hand. In other words, mirror neurons first evolved to augment the "canonical" F5 neurons (active during self-movement based on observation of an object) by providing visual feedback on "hand state," relating the shape of the hand to the shape of the object. We then introduce the MNS1 (mirror neuron system 1) model of F5 and related brain regions. The existing Fagg–Arbib–Rizzolatti–Sakata model represents circuitry for visually guided grasping of objects, linking the anterior intraparietal area (AIP) with F5 canonical neurons. The MNS1 model extends the AIP visual pathway by also modeling pathways, directed toward F5 mirror neurons, which match arm–hand trajectories to the affordances and location of a potential target object. We present the basic schemas for the MNS1 model, then aggregate them into three "grand schemas" – visual analysis of hand state, reach and grasp, and the core mirror circuit – for each of which we present a useful implementation (a non-neural visual processing system, a multijoint 3-D kinematics simulator, and a learning neural network, respectively). With this implementation we show how the mirror system may *learn* to recognize actions already in the repertoire of the F5 canonical neurons. We show that the connectivity pattern of mirror neuron circuitry can be established through training, and that the resultant network can exhibit a range of novel, physiologically interesting behaviors during the process of action recognition. We train the system on the basis of final grasp but then observe the whole time course of mirror neuron activity, yielding predictions for neurophysiological experiments under conditions of spatial perturbation, altered kinematics, and ambiguous grasp execution which highlight the importance of the *timing* of mirror neuron activity.

Correspondence to: M.A. Arbib (e-mail: arbib@pollux.usc.edu)

1 Introduction

1.1 The mirror neuron system for grasping

The macaque inferior premotor cortex has been identified as being involved in reaching and grasping movements (Rizzolatti et al. 1988). This region has been further partitioned into two subregions: F5, the rostral region, located along the arcuate part; and F4, the caudal part (see Fig. 1). The neurons in F4 appear to be primarily involved in the control of proximal movements (Gentilucci et al. 1988), whereas the neurons of F5 are involved in distal control (Rizzolatti et al. 1988).

Rizzolatti et al. (1996a) and Gallese et al. (1996) discovered a subset of F5 hand cells, which they called *mirror neurons*. Like other F5 neurons, mirror neurons are active when the monkey performs a particular class of actions, such as grasping, manipulating, and placing. However, in addition the mirror neurons become active when the monkey observes the experimenter or another monkey performing an action. The term F5 *canonical neurons* is used to distinguish the F5 hand cells which do *not* posses the mirror property but are instead responsive to visual input concerning a suitably graspable object. The canonical neurons are indistinguishable from the mirror neurons with respect to their firing during self-action. However they are different in their visual properties – they respond to object presentation and not action observation per se (Murata et al. 1997).

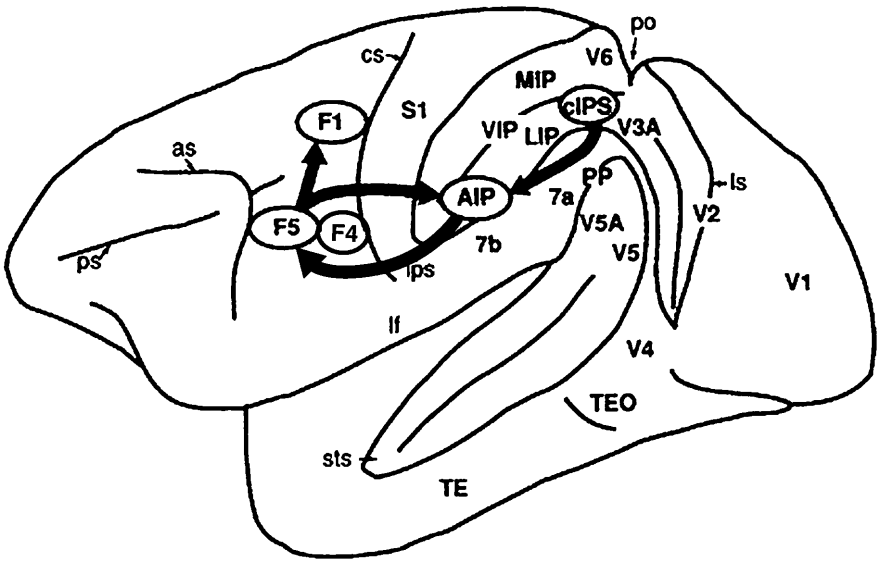

Fig. 1. Lateral view of the monkey cerebral cortex (intraparietal sulcus, *IPS*; superior temporal sulcus, *STS*; and lunate sulcus opened). The visuomotor stream for hand action is indicated by *arrows* (adapted from Sakata et al. 1997)

Most mirror neurons exhibit a clear relation between the observed and executed actions for which they are active. The congruence between the observed and executed action varies. For some of the mirror neurons, the congruence is quite loose; for others, not only must the general action (e.g., grasping) match, but also the way the action is executed (e.g., power grasp) must match as well. To be triggered, the mirror neurons require an interaction between the hand motion and the object. The vision of the hand motion or the object alone does not trigger mirror activity (Gallese et al. 1996).

It has thus been argued that the importance of mirror neurons is that they provide a neural representation that is common to execution and observation of grasping actions and thus that these neurons are crucial to the social interactions of monkeys, providing the basis for *understanding* of actions by others through their linkage of action and perception (Rizzolatti and Fadiga 1998). Below we offer the hand-state hypothesis, suggesting that this important role is an exaptation of a more primitive role, namely that of providing feedback for visually guided grasping movements. By *exaptation* we mean the exploitation of an adaptation of a system to serve a different purpose (in this case for social understanding) than it initially developed for (in this case, visual control of grasping). We will then develop the MNS1 (mirror neuron system 1) model and show that the system can exploit its ability to relate self-hand movements to objects to recognize the manual actions being performed by others, thus yielding the mirror property. We also conduct a number of simulation experiments with the model and show that these yield novel predictions, suggesting new neurophysiological experiments to further probe the monkey mirror system. However, before introducing the hand-state hypothesis and the MNS1 model, we first outline the FARS (Fagg–Arbib–Rizzolatti–Sakata) model of the circuitry that includes the F5 canonical neurons and provides the conceptual basis for the MNS1 model.

1.2 The FARS model of parietal–premotor interactions in grasping

Studies of the anterior intraparietal sulcus (AIP; Fig. 1) revealed cells that were activated by the sight of objects for manipulation (Taira et al. 1990; Sakata et al. 1995). In addition, this region has very significant recurrent corticocortical projections with area F5 (Matelli et al. 1994; Sakata et al. 1997). In their computational model for primate control of grasping (the FARS model), Fagg and Arbib (1998) analyzed these findings of Sakata and Rizzolatti to show how F5 and AIP may act as part of a visuomotor transformation circuit, which carries the brain from sight of an object to the execution of a particular grasp. In developing the FARS model, Fagg and Arbib (1998) interpreted the findings of Sakata (on AIP) and Rizzolatti (on F5) as showing that AIP represents the grasps *afforded* by the object while F5 selects and drives the execution of the grasp. The term *affordance* (adapted from Gibson 1966) refers to parameters for motor interaction that are signaled by sensory cues without invocation of high-level object recognition processes. The model also suggests how F5 may use task information and other constraints encoded in prefrontal cortex (PFC) to resolve the action opportunities provided by multiple affordances. Here we emphasize the essential components of the model (Fig. 2) that will ground the version of the MNS1 model presented below. We focus on the linkage between viewing an affordance of an object and the generation of a single grasp:

1. The dorsal visual stream (parietal cortex) extracts parametric information about the object being attended. It does not "know" what the object is; it can only see the object as a set of possible affordances. The ventral stream (from primary visual cortex to inferotemporal cortex, IT), by contrast, recognize what the object is and passes this information to PFC which can then, on the basis of the current goals of the organism and the recognition of the nature of the

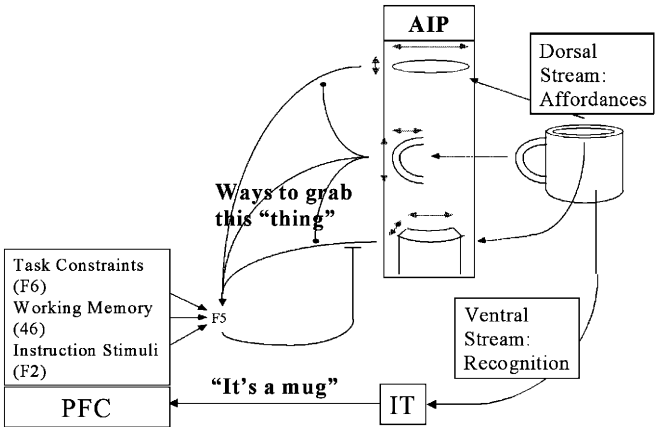

Fig. 2. The anterior intraparietal area, (*AIP*) extracts the affordances and F5 selects the appropriate grasp from the AIP "menu." Various biases are sent to F5 by the prefrontal cortex (*PFC*) which relies on the recognition of the object by the inferotemporal cortex (*IT*). The dorsal stream through AIP to F5 is replicated in the current version of the mirror neuron system 1 (MNS1) model; the influence of IT and PFC on F5 is not analyzed further in the present paper

object, bias F5 to choose the affordance appropriate to the task at hand.

2. AIP is hypothesized as playing a dual role in the seeing/reaching/grasping process, not only computing affordances exhibited by the object but also, as one of these affordances is selected and execution of the grasp begins, serving as an active memory of the one selected affordance and updating this memory to correspond to the grasp that is actually executed.
3. F5 is hypothesized as first being responsible for integrating task constraints with the set of grasps that are afforded by the attended object in order to select a single grasp. After selection of a single grasp, F5 unfolds this represented grasp in time to govern the role of primary motor cortex (F1) in its execution.
4. In addition, the FARS model represents the way in which F5 may accept signals from areas F6 (pre-SMA supplementary motor area), 46 (dorsolateral prefrontal cortex), and F2 (dorsal premotor cortex) to respond to task constraints, working memory, and instruction stimuli, respectively, and how these in turn may be influenced by object recognition processes in IT (see Fagg and Arbib 1988 for more details), but these aspects of the FARS model are not involved in the current version of the MNS1 model.

2 The hand-state hypothesis

The key notion of the MNS1 model is that the brain augments the mechanisms modeled by the FARS model – for recognizing the grasping-affordances of an object (AIP) and transforming these into a program of action – by mechanisms that can recognize an action in terms of the hand state which makes explicit the relation between the unfolding trajectory of a hand and the affordances of an object. Our radical departure from all prior studies of the mirror system is to hypothesize that this system evolved in the first place to provide feedback for visually directed grasping, with the social role of the mirror system being an exaptation as the hand-state mechanisms become applied to the hands of others as well as to the hand of the animal itself.

2.1 Virtual fingers

As background for the hand-state hypothesis, we first present a conceptual analysis of grasping. Iberall and Arbib (1990) introduced the theory of *virtual fingers* and *opposition space*. The term *virtual finger* is used to describe the physical entity (e.g., one or more fingers, and the palm of the hand) that is used in applying force, and thus includes specification of the region to be brought in contact with the object (what we might call the "virtual fingertip"). Figure 3 shows three types of opposition: those for the precision grip, power grasp, and side opposition. Each of the grasp types is defined by specifying two virtual fingers, VF1 and VF2, and the regions on VF1 and VF2 which are to be brought into contact with the object to grasp it. Note that the "virtual fingertip" for VF1 in palm opposition is the surface of the palm, while that for VF2 in side opposition is the side of the index finger. The grasp defines two "opposition axes": the *opposition axis in the hand* joining the virtual finger regions to be opposed to each other, and the *opposition axis in the object* joining the regions where the virtual fingers contact the object. Visual perception provides *affordances* (different ways to grasp the object); once an affordance is selected, an appropriate opposition axis in the object can be determined. The task of motor control is to preshape the hand to form an opposition axis appropriate to the chosen affordance, and to move the arm so as to transport the hand to bring the hand and object axes into alignment. During the last stage of transport, the virtual fingers move down the opposition axis (the "enclose" phase) to grasp the object just as the hand reaches the appropriate position.

2.2 The hand-state hypothesis

We assert as a general principle of motor control that if a motor plant is used for a task, then a feedback system will evolve to better control its performance in the face of perturbations. We thus ask, as a sequel to the work of Iberall and Arbib (1990), what information would be needed by a feedback controller to control grasping in the manner described in Sect. 2.1. Modeling of this feedback control is beyond the scope of this paper. Rather, our aim is to show how the availability of such feedback signals in the primate cortex for self-action for manual grasping can provide the action recognition capabilities which characterize the mirror system. Specifically, we offer the following hypothesis.

The hand-state hypothesis. The basic functionality of the F5 mirror system is to elaborate the appropriate feedback – what we call the *hand state* – for opposition-space-based control of manual grasping of an object.

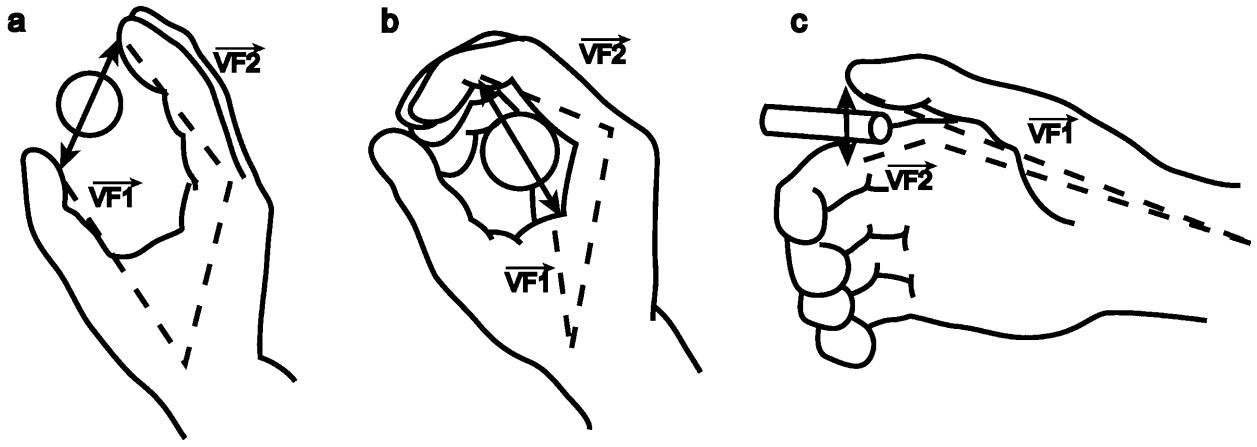

Fig. 3a–c. Each of the three grasp types here is defined by specifying two "virtual fingers", *VF1* and *VF2*, which are groups of fingers or a part of the hand such as the palm which are brought to bear on either side of an object to grasp it: **a** pad oppostion; **b** palm opposition; **c** side opposition. The specification of the virtual fingers includes specification of the region on each virtual finger to be brought in contact with the object. A successful grasp involves the alignment of two "opposition axes": the *opposition axis in the hand* joining the virtual finger regions to be opposed to each other, and the *opposition axis in the object* joining the regions where the virtual fingers contact the object (Iberall and Arbib 1990)

Given this functionality, the social role of the F5 mirror system in understanding the actions of others may be seen as an exaptation gained by generalizing from one's own hand to another's hand.

The key to the MNS1 model, then, is the notion of *hand state* as encompassing data required to determine whether the motion and preshape of a moving hand may be extrapolated to culminate in a grasp appropriate to one of the affordances of the observed object. Basically a mirror neuron must fire if the preshaping of the hand conforms to the grasp type with which the neuron is associated; and the extrapolation of *hand state* yields a time at which the hand is grasping the object along an axis for which that affordance is appropriate.

Our current representation of hand state defines a seven-dimensional trajectory

$$\boldsymbol{F}(t) = [d(t), v(t), a(t), o_1(t), o_2(t), o_3(t), o_4(t)]$$

with the following components (see Fig. 4): three components are hand configuration parameters—$a(t)$, index finger-tip and thumb-tip aperture; $o_3(t)$, $o_4(t)$, the two angles defining how close the thumb is to the hand as measured relative to the side of the hand and to the inner surface of the palm—and the remaining four parameters relate the hand to the object. o_1 and o_2 components represent the orientation of different components of the hand relative to the opposition axis for the chosen affordance in the object, whereas d and v represents the kinematics properties of the hand with reference to the target location—$o_1(t)$, the cosine of the angle between the object axis and the (*index finger tip–thumb tip*) vector; $o_2(t)$, the cosine of the angle between the object axis and the (*index finger knuckle–thumb tip*) vector; $d(t)$, distance to target at time t; $v(t)$, tangential velocity of the wrist.

In considering the last four variables, note that only one or two of them will be relevant in generating a specific type of grasp, but they all must be available to monitor a wide range of possible grasps. We have chosen a set of variables of clear utility in monitoring the successful progress of grasping an object, but do not claim that these and only these variables are represented in the brain. Indeed, the brain's actual representation will be a distributed neural code, which we predict will correlate with such variables, but will not be decomposable into a coordinate-by-coordinate encoding. However, we believe that the explicit definition of hand state offered here will provide a firm foundation for the design of new experiments in kinesiology and neurophysiology.

The crucial point is that the availability of the hand state to provide feedback for visually directed grasping makes action recognition possible. Notice that we have carefully defined the hand state in terms of relationships between hand and object (though the form of the definition must be subject to future research). This has the benefit that it will work just as well for measuring how the monkey's own hand is moving to grasp an object as for observing how well another monkey's hand is moving to grasp the object. This, we claim, is what allows self-observation by the monkey to train a system that can be used for observing the actions of others and recognizing just what those actions are.

3 The MNS1 model

We now present a high-level view of the MNS1 model in terms of the set of interacting *schemas* (functional units: Arbib 1981; Arbib et al. 1998, Chap. 3) shown in Fig. 5, which define the MNS1 model of area F5 and related brain regions. As we now demonstrate, the connectivity shown in Fig. 5 is constrained by the existing neurophysiology and neuroanatomy of the monkey brain. We have already introduced areas AIP and area F5, dividing the F5 grasp-related neurons into (i) F5 *mirror neurons*

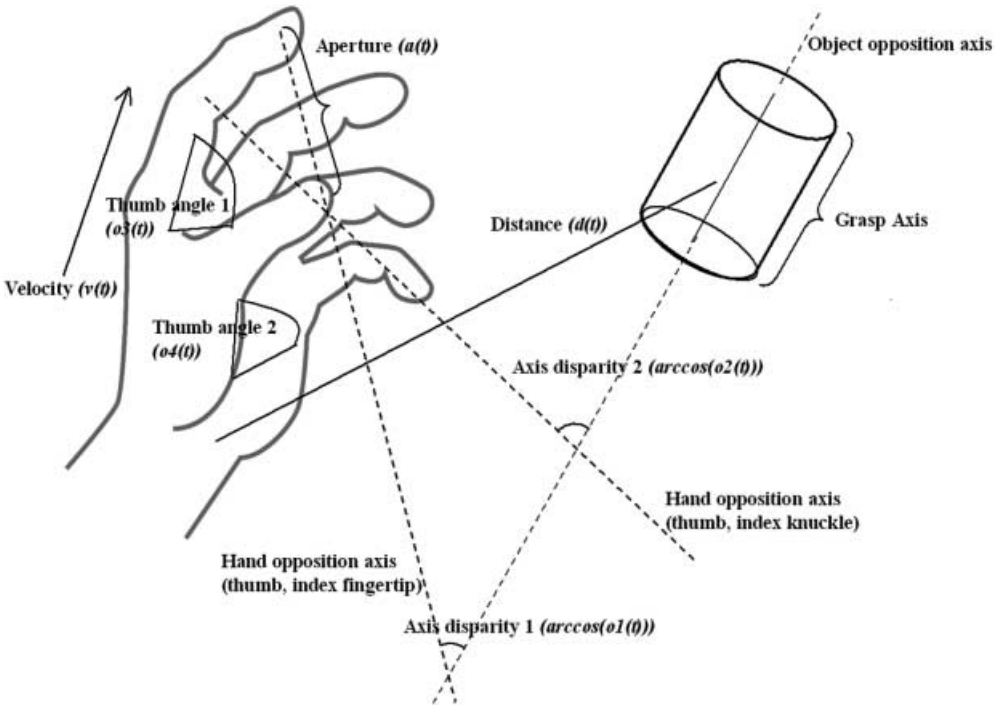

Fig. 4. The components of hand state $F(t) = [d(t), v(t), a(t), o_1(t), o_2(t), o_3(t), o_4(t)]$. Note that some of the components are purely hand configuration parameters (namely v, o_3, o_4, and a) whereas others are parameters relating the hand to the object

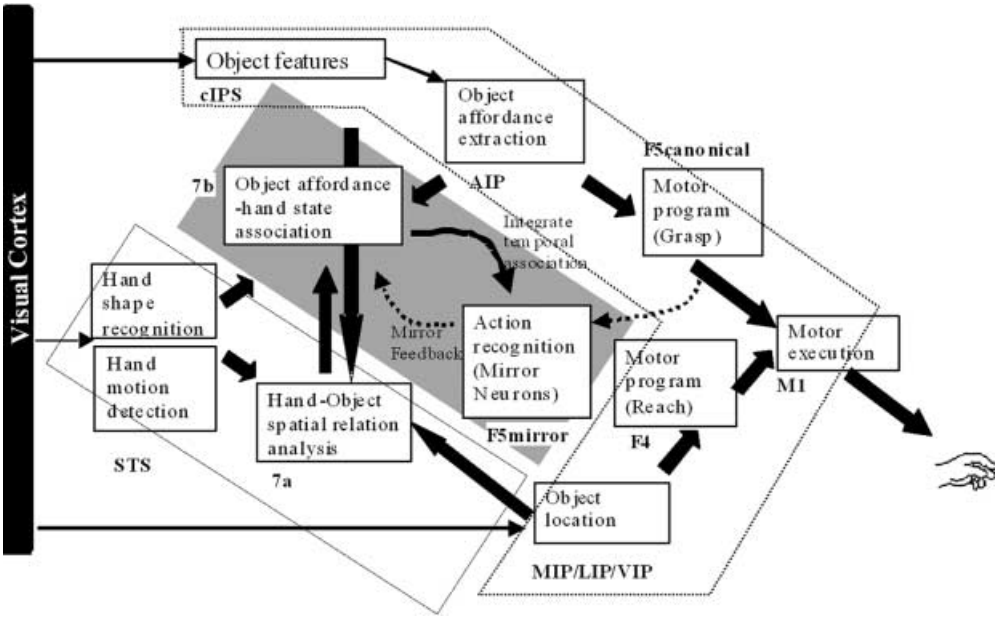

Fig. 5. The MNS1 model. *Top diagonal*: a portion of the FARS model. Object features are processed by the caudal intraparietal sulcus, (c*IPS*) and *AIP* to extract grasp affordances, and these are sent on to the canonical neurons of *F5* that choose a particular grasp. *Bottom right*: recognizing the location of the object provides parameters to the motor programming area *F4* which computes the reach. The information about the reach and the grasp is taken by the motor cortex *M1* to control the hand and the arm. New elements of the MNS1 model: *bottom left* are two schemas, one to recognize the shape of the hand, and the other to recognize how that hand is moving. Just to the right of these is the schema for hand–object spatial relation analysis. It takes information about object features, the motion of the hand and the location of the object to infer the relation between hand and object. The *center two regions* marked by the *gray rectangle* form the core mirror circuit. This complex associates the visually derived input (hand state) with the motor program input from *F5 canonical* neurons during the learning process for the mirror neurons. The grand schemas introduced in section 3.2 are illustrated as the following. The "core mirror circuit" schema is marked by the *center grey box*; the "visual analysis of hand state" schema is outlined by *solid lines* just below it, and the "reach and grasp" schema is outlined by *dashed lines* (*Solid arrows*: established connections; *Dashed arrows*: postulated connections. Details of the ascription of specific schemas to specific brain regions were deferred to a later paper.)

which are, when fully developed, active during certain self-movements of grasping by the monkey and during the observation of a similar grasp executed by others, and (ii) F5 *canonical neurons*, namely those active during self-movement and object vision but not for recognition of the action of others. Other brain regions also play an

important role in mirror-neuron-system functioning in the macaque's brain. Posterior parietal cortex and the cortex of caudal superior temporal sulcus (STS) have been subdivided into numerous areas mainly involved in spatial analysis of the visual environment and in the control of spatially oriented behaviour (Maioli et al. 1998).

Based on cytoarchitectonic and connectional criteria Brodmann's area 7 (inferior parietal lobule) has been found to contain distinct regions including areas 7a and 7b and area 7ip. Area 7 reaches its highest development in primates (Cavada and Goldman-Rakic 1989). Damage to this area can cause impairments in visually guided reaching in addition to other spatial perceptual and motor deficits (Ratcliff 1991; Stein 1991). The posterior half of 7ip corresponds to the ventral intraparietal area (Maunsell and Van Essen 1983) and the lateral intraparietal area (Andersen et al. 1985), and contains area AIP. Areas 7a, 7ip, and 7b (to a lesser extent) are reciprocally connected with the cortex of the STS. Although the density of 7b connections with the visual motion cortex of STS is weak compared the extensive connections of 7b with somatosensory areas, the interconnections of 7b with the visual regions are established through anterior 7ip and the transitional cortex 7ab between 7a and 7b (Cavada and Goldman-Rakic 1989). Findings from the same study also confirm that AIP is connected with area 7b.

Area F5 has no direct input from visual occipital areas. Its main cortical input comes from inferior parietal lobe, and in particular area AIP and area 7b (Matelli et al. 1985; Gallese et al. 1996). The detailed connections and organization of areas F4 and F5 are reviewed by Rizzolatti et al. (1998) and Geyer et al. (2000). The caudal intraparietal sulcus, with projections to AIP (Sakata et al. 1997) and area 7a (Cavada and Goldman-Rakic 1989, Fig. 7), has been shown to encode object properties such as the orientation of object surfaces and object axes (Sakata et al. 1997).

We conclude our brief discussion of anatomical connections by turning back to area 7ip. Area 7a receives input from the lateral intrparietal area (Andersen et al. 1990; Lewis and Van Essen 2000), the medial intraparietal area (Boussaoud et al. 1990; Lewis and Van Essen 2000; Bota 2001), and the ventral intraparietal area (Lewis and Van Essen 2000; reviewed in Maunsell 1995). Interested readers can find more details about these connections at the NeuroHomology Database Website (http://brancusi.usc.edu/scripts/webmerger.exe?/database/homologies-main.html; Bota 2001). The premotor projections of these areas include the regions F2 (not shown) and F4 (Luppino et al. 1999), as reviewed in Geyer et al. (2000).

The subsystem of the MNS1 model responsible for the visuomotor transformation of objects into affordances and grasp configurations, linking AIP and F5 canonical neurons, corresponds to a key subsystem of the FARS model reviewed above. Our task is to complement the visual pathway via AIP by pathways directed toward F5 mirror neurons which allow the monkey to observe arm–hand trajectories and match them to the affordances and location of a potential target object. We will then show how the mirror system may *learn* to recognize actions already in the repertoire of the F5 canonical neurons. In short, we will provide a mechanism whereby the actions of others are ''recognized'' based on the circuitry involved in performing such actions. (A topic of future research is to delineate the circuitry – more apparent in chimpanzee than monkey but still rudimentary compared to human – for the reverse process of imitation, going from the recognition of a novel action performed by others to the addition of that action to one's own repertoire.) Sect. 4 provides the details of the implemented schemas, and Sect. 5 confronts the overall model with virtual experiments and produces testable predictions.

3.1 Overall function

In general, the visual input to the monkey represents a complex scene. However, we here sidestep much of this complexity (including attentional mechanisms) by assuming that the brain extracts two salient subscenes; a stationary object and in some cases a (possibly) moving hand. The overall system operates in two modes:

1. *Prehension*. In this mode, the view of the stationary object is analyzed to extract affordances; then under prefrontal influence F5 may choose one of these to act upon, commanding the motor apparatus to perform the appropriate reach and grasp based on parameters supplied by the parietal cortex. The FARS model captures the linkage of F5 and AIP with PFC (Fig. 2). In the MNS1 model, we incorporate the F5 and AIP components from FARS (top diagonal of schemas in Fig. 5), but omit IT and PFC from the present analysis.
2. *Action recognition*. In this mode, the view of the stationary object is again analyzed to extract affordances, but now the initial trajectory and preshape of an observed moving hand must be extrapolated to determine whether the current motion of the hand can be expected to culminate in a grasp of the object appropriate to one of its affordances.

We do not prespecify all the details of the MNS1 schemas. Instead, we offer a learning model which, given a grasp that is already in the motor repertoire of the F5 canonical neurons, can yield a set of F5 mirror neurons trained to be active during such grasps as a result of *self-observation* of the monkey's own hand grasping the target object. (How such grasps may be acquired in the first place is a topic of current research.) Consistent with the hand-state hypothesis, the result will be a system whose mirror neurons can respond to *similar actions observed being performed by others*. The current implementation of the MNS1 model exploits learning in artificial neural nets.

The heart of the learning model is provided by the *object affordance–hand state association* schema and the a*ction recognition (mirror neurons)* schema. These form the *core mirror (learning) circuit*, marked by the gray

slanted rectangle in Fig. 5, which mediates the development of mirror neurons via learning. The simulation results of this article will focus on this part of the model. Section 4 presents in detail the neural network structure of the core circuit. As we note further in Sect. 5, this leaves open many problems for further research, including the development of a basic action repertoire by F5 canonical neurons through trial and error in infancy, and the expansion and refinement of this repertoire throughout life.

3.2 Schemas explained

As shown in the caption of Fig. 5, we encapsulate the schemas shown there into the three "grand schemas" of Fig. 6a. These guide our implementation of MNS1. Nonetheless, it seems worth providing specifications of the more detailed schemas both to ground the definition of the grand schemas and to set the stage for more detailed neurobiological modeling in later papers. Our earlier review of the neuroscience literature justifies our initial hypotheses, made explicit in Fig. 5, as to where these finer-grain schemas are realized in the monkey brain. However, after we explain these finer-grain schemas, we will then turn to our present simulation of the three grand schemas which is based on overall functionality rather than neural regionalization, yet nonetheless yields interesting predictions for further neurophysiological experimentation.

3.2.1 Grand schema 1: reach and grasp

Object features schema. The output of this schema provides a coarse coding of geometrical features of the observed object. It thus provides suitable input to AIP and other regions/schemas.

Object affordance extraction schema. This schema transforms its input, the coarse coding of geometrical features of the observed object provided by the o*bject features* schema, into a coarse coding for each affordance of the observed object.

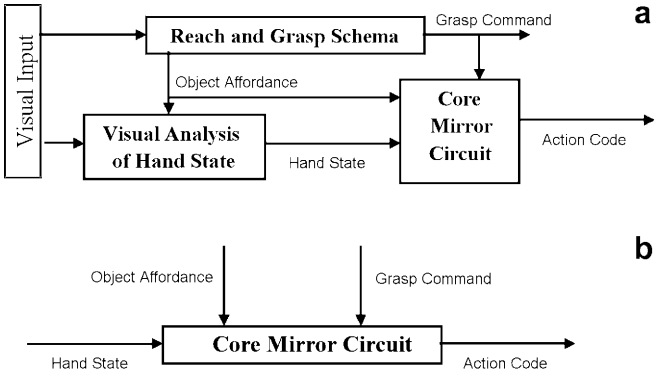

Fig. 6a,b. **a** For purposes of simulation, we aggregate the schemas of the MNS1 model of Fig. 5 into three "grand schemas" for *visual analysis of hand state*, *reach and grasp*, and *core mirror circuit*. **b** For detailed analysis of the core mirror circuit, we dispense with simulation of the other two grand schemas and use other computational means to provide the three key inputs to this grand schema

Motor program (grasp) schema. We identify this schema with the canonical F5 neurons, as in the FARS model. Input is provided by AIP's coarse coding of affordances for the observed object. We assume that the output of the schema encodes a generic motor program for the AIP-coded affordances. This output serves as the learning signal to the *action recognition* (Mirror neurons) schema and drives the hand control functions of the *motor execution* schema.

Object location schema. The output of this schema provides, in some body-centered coordinate frame, the location of the center of the opposition axis for the chosen affordance of the observed object.

Motor program (reach) schema. The input is the position coded by the *object location* schema, while the output is the motor command required to transport the arm to bring the hand to the indicated location. This drives the arm control functions of the *motor execution* schema.

The *motor execution schema* determines the course of movements via activity in primary motor cortex M1 and "lower" regions.

We next review the schemas which (in addition to the previously presented *object features* and *object affordance extraction* schemas) implement the visual system of the model.

3.2.2 Grand schema 2: visual analysis of hand state. The *hand-shape recognition schema* takes as input a view of a hand, and its output is a specification of the hand shape, which thus forms some of the components of the hand state. In the current implementation these are $a(t)$, $o_3(t)$, and $o_4(t)$. Note also that we implicitly assume that the schema includes a validity check to verify that the scene does contain a hand.

The *Hand-motion detection schema* takes as input a *sequence* of views of a hand and returns as output the wrist velocity, supplying the v(t) component of the hand state.

The *Hand-object spatial relation analysis schema* receives object-related signals from the *object features* schema, as well as input from the *object location*, *hand-shape recognition,* and *Hand-motion detection* schemas. Its output is a set of vectors relating the current hand preshape to a selected affordance of the object. The schema computes such parameters as the distance of the object to the hand, and the disparity between the opposition axes of the object and the hand. Thus the hand-state components $o_1(t)$, $o_2(t)$, and $d(t)$ are supplied by this schema. The *hand–object spatial relation analysis* schema is needed because, for almost all mirror neurons in the monkey, a hand mimicking a matching grasp would fail to elicit the mirror neuron's activity unless the hand's trajectory were taking it toward an object with a grasp that matches one of the affordances of the object. The output of this visual analysis is relayed to the *object affordance–hand state association* schema which drives the F5 mirror neurons whose output is a signal expressing confidence that the observed trajectory will extrapolate to match the observed target object using the grasp encoded by that mirror neuron.

3.2.3 Grand schema 3: core mirror circuit. The *action recognition schema* – which is meant to correspond to the mirror neurons of area F5 – receives two inputs in our model. One is the motor program selected by the *motor program* schema; the other comes from the *object affordance–hand state association* schema. This schema works in two modes: learning and recognition. When a self-executed grasp is taking place the schema is in learning mode and the association between the observed hand state (o*bject affordance–hand state association* schema*)* and the motor program (*motor program* schema) is learned. While in recognition mode, the motor program input is not active and the schema acts as a recognition circuit. If satisfactory learning (in terms of generalization and the range of actions learned) has taken place via self-observation, then the schema will respond correctly while observing another's grasp actions.

The *object affordance–hand state association schema* combines all the hand-related information as well as the object information available. Thus the inputs to the schema are from *hand-shape recognition* [components $a(t)$, $o_3(t)$, $o_4(t)$], *hand motion detection* [component $v(t)$], *hand–object spatial relation analysis* [$o_1(t)$, $o_2(t)$, $d(t)$] and from *object affordance extraction* schemas. As will be explained below, the schema needs a learning signal (mirror feedback). This signal is relayed by the *action recognition* schema, and is basically a copy of the motor program passed to the *action recognition* schema itself. The output of this schema is a distributed representation of the object and hand-state match (in our implementation the representation is not prespecified but shaped by the learning process). The idea is to match the object and the hand state as the action progresses during a specific observed reach and grasp. In the current implementation, time is unfolded into a spatial representation of "the trajectory until now" at the input of the *object affordance–hand state association* schema, and the action recognition schema decodes the distributed representation to form the mirror response (again, the decoding is not prespecified but is the result of the backpropagation learning). In any case, the schema has two operating modes. The first is the learning mode where the schema tries to adjust its efferent and afferent weights to ensure the right activity in the *action recognition* schema. The second mode is the forward mode where it maps the hand state and the object affordance into a distributed representation to be used by the *action recognition* schema.

The key question for our present modeling will be to account for how learning mechanisms may shape the connections to mirror neurons in such a way that an action in the motor program repertoire of the F5 canonical neurons may become recognized by the mirror neurons when performed by others.

To conclude this section, we note that our modeling is subject to two quite different tests: (i) its overall efficacy in explaining behavior and its development, which can be tested at the level of the schemas (functional units) presented in this article; and (ii) its further efficacy in explaining and predicting neurophysiological data. As we shall see below, certain neurophysiological predictions are possible given the current work, even though the present implementation relies on relatively abstract artificial neural networks.

4 Methods

4.1 Schema implementation

Having indicated the functionality and possible neural basis for each of the schemas that will make up each grand schema, we now turn to the implementation of these three grand schemas. As noted earlier, the detailed neurobiological modeling of the finer-grain schemas is a topic for further research. Here we implement the three grand schemas so that each functions correctly in terms of its input–output relations, and so that the core mirror circuit contains model neurons whose behavior can be tested against neurophysiological data and yield predictions for novel neurophysiological experiments. The core mirror circuit is thus the principal object of study (Fig. 6b), but in order to study it there must be an appropriate context, necessitating the construction of the kinematically realistic reach and grasp simulator and the visual analyzer for hand state. The latter will first be implemented as an analyzer of views of human hands, and then will have its output replaced by simulated hand-state trajectories to reduce computational expense in our detailed analysis of the core mirror circuit.

4.2 Grand schema 1: reach and grasp

We first discuss the reach and grasp simulator that corresponds to the whole reach and grasp command system shown at the right of the MNS1 diagram (Fig. 5). The simulator lets us move from the representation of the shape and position of a (virtual) 3-D object and the initial position of the (virtual) arm and hand to a trajectory that successfully results in simulated grasping of the object. In other words the simulator plans a grasp and reach trajectory and executes it in a simulated 3-D world. Trajectory planning (e.g., Kawato et al. 1987; Jordan and Rumelhart 1992; Kawato and Gomi 1992; Karniel and Inbar 1997; Breteler et al. 2001) and control of prehension (Hoff and Arbib 1993; Wolpert and Ghahramani 2000, for a review), and their adaptation, have been widely studied. However, our simulator is not adaptive – its sole purpose is to create kinematically realistic actions. A similar reach and grasp system was proposed (Rosenbaum et al. 1999) where a movement is planned based on the constraint hierarchy, relying on obstacle avoidance and candidate posture evaluation processes (Meulenbroek et al. 2001). However, the arm and hand model was much simpler than ours as the arm was modeled as a 2-D kinematics chain. Our reach/grasp simulator is a nonneural extension of FARS model functionality to include the reach component. It controls a virtual arm/hand with 19 degrees of freedom (DOFs) (three at the shoulder, one for elbow flexion/extension,

three for wrist rotation, two for each finger joints with additional two DOFs for thumb one to allow the thumb to move sideways, and the other for the last joint in the thumb) and provides routines to perform realistic grasps. This kinematic realism is based on the literature of primate reach and grasp experiments [for human see Jeannerod et al. (1995), Hoff and Arbib (1993) and citations therein; for monkey see Roy et al. (2000)]. During a typical reach to grasp movement, the hand will follow a bell-shaped velocity profile (a single peaked velocity curve). The kinematics of the aperture between fingers used for grasping also exhibits typical characteristics. The aperture will first reach a maximum value that is larger than the aperture required for grasping the object, and then as the hand approaches to the target the hand encloses to match the actual required aperture for the object. It is also important to note that in grasping tasks the temporal pattern of reaching and grasping is similar in monkey and human (Roy et al. 2000). Of course, there are intersubject and intertrial variability in both velocity and aperture profiles (Marteniuk and MacKenzie 1990). Therefore in our simulator we captured the qualitative aspects of the typical reach and grasp actions, namely that the velocity profiles have single peaks and that the hand aperture has a maximum value which is larger than the object size (see Fig. 7, curves $a(t)$ and $v(t)$, for sample aperture and velocity profiles generated by our simulator) . A grasp is planned by first setting the operational space constraints (e.g., points of contact of fingers on the object) and then finding the arm–hand configuration to fulfill the constraints. The latter is the inverse kinematics problem. The simulator solves the inverse kinematics problem by simulated gradient descent with noise added to the gradient (see Appendix A.2 for a grasp-planning example). Once the hand–arm configuration is determined for a grasp action, then the trajectory is generated by warping time using a cubic spline. The parameters of the spline are fixed and determined empirically to satisfy aperture and velocity profile requirements. Within the simulator, it is possible to adjust the target identity, position and size manually using a graphical user interface or automatically by the simulator as, for example, in training set generation.

Figure 7 (left) shows the end state of a power grasp, while Fig. 7 (right) shows the time series for the hand state associated with this simulated power grasp trajectory. For example, the curve labeled $d(t)$ shows the distance from the hand to the object decreasing until the grasp is completed, while the curve labeled $a(t)$ shows how the aperture of the hand first increases to yield a safety margin larger than the size of the object and then decreases until the hand contacts the object.

Figure 8a shows the virtual hand/arm holding a small cube in a *precision grip* in which the index finger (or a larger "virtual finger") opposes the thumb. The power grasp (Fig. 8b) is usually applied to big objects and is characterized by the hands covering the object, with the fingers as one virtual finger opposing the palm as the other. In a side grasp (Fig. 8c), the thumb opposes the side of another finger. To clarify the type of heuristics we use to generate the grasp, Appendix A.2 outlines the grasp planning and execution for a precision pinch.

4.3 Grand schema 2: visual analysis of hand state

The visual analysis of hand state schema is a nonneurophysiological implementation of a visual analysis system to validate the extraction of hand parameters from a view of a hand, by recovering the configuration of a model of the hand being seen. The hand model is a 3-D 14 DOF kinematic model, with a 3-DOF joint for the wrist, two 1-DOF joints (metacarpophalangeal and distalinterphalangeal) for each of the four fingers, and finally a 1-DOF joint for the metacarpophalangeal joint and a 2-DOF joint for the carpometacarpal joint of the

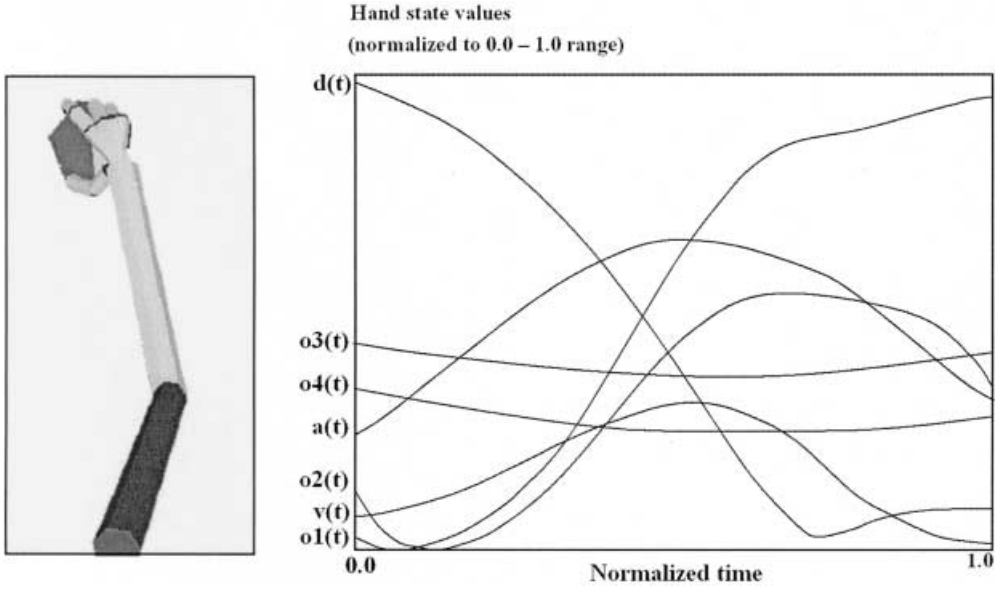

Fig. 7. *Left*: The final state of arm and hand achieved by the reach/grasp simulator in executing a power grasp on the object shown. *Right*: The hand-state trajectory read off from the simulated arm and hand during the movement whose end state is shown at left. The hand state components are: $d(t)$, distance to target at time t; $v(t)$, tangential velocity of the wrist; $a(t)$, index and thumb finger tip aperture; $o_1(t)$, cosine of the angle between the object axis and the (index finger tip–thumb tip) vector; $o_2(t)$, cosine of the angle between the object axis and the (index finger knuckle—thumb tip) vector; $o_3(t)$, angle between the thumb and the palm plane; $o_4(t)$, angle between the thumb and the index finger

thumb. Note the distinction between "hand configuration," which gives the joint angles of the hand considered in isolation, and the "hand state" which comprises seven parameters relevant to assessing the motion and preshaping of the hand relative to an object. Thus the hand configuration provides some–but not all–of the data needed to compute the hand state.

To lighten the load of building a visual system to recognize hand features, we mark the wrist and the articulation points of the hand with distinctive colors. We then use this color coding to help recognize key portions of the hand and use this result to initiate a process of model matching. Thus the first step of the vision problem is color segmentation, after which the 3-D hand shape is recovered.

4.3.1 Color segmentation and feature extraction. One needs *color segmentation* to locate the colored regions on the image. Gray-level segmentation techniques cannot be used in a straightforward way because of the vectorial nature of color images (Lambert and Carron 1999). Split-and-merge is a well-known image segmentation technique in image processing (Sonka et al. 1993), which involves recursively splitting the image into smaller pieces until some homogeneity criterion is satisfied as a basis for reaggregation into regions. In our case, the criterion is having similar color throughout a region. However, RGB (red–green–blue) space is not well suited for this purpose; HSV (hue–saturation–value) space is better suited since in the segmentation processes hue usually corresponds to human perception and it ignores shading effects (Russ 1998, Chaps. 1, 6). However, the segmentation system we implemented with HSV space, although better than the RGB version, was not satisfactory for our purposes. Therefore, we designed a system that can learn the best color space.

Figure 9a shows the training phase of the *color expert* system, which is a (one-hidden-layer) feed-forward network with a sigmoidal activation function. The learning algorithm is backpropagation with momentum and an adaptive learning rate. The given image is put through a smoothing filter to reduce noise in the image before training. Then the network is given around 100 training samples each of which is a vector of [(R, G, B), *perceived color code*] values. The output color code is a vector consisting of all zeros except for one component corresponding to the *perceived* color of the patch. Basically, the training builds an internal nonlinear color space from which it can unambiguously tell the perceived color. This training is done only at the beginning of a session to learn the colors used on the particular hand. Then the network is fixed as the hand is viewed in a variety of poses.

Figure 9b illustrates the actual segmentation process using the color expert to find each region of a single (perceived) color (see Appendix A.1 for details). The output of the algorithm is then converted into a feature

Fig. 8a–c. Grasps generated by the simulator: **a** a precision grasp; **b** a power grasp; **c** a side grasp

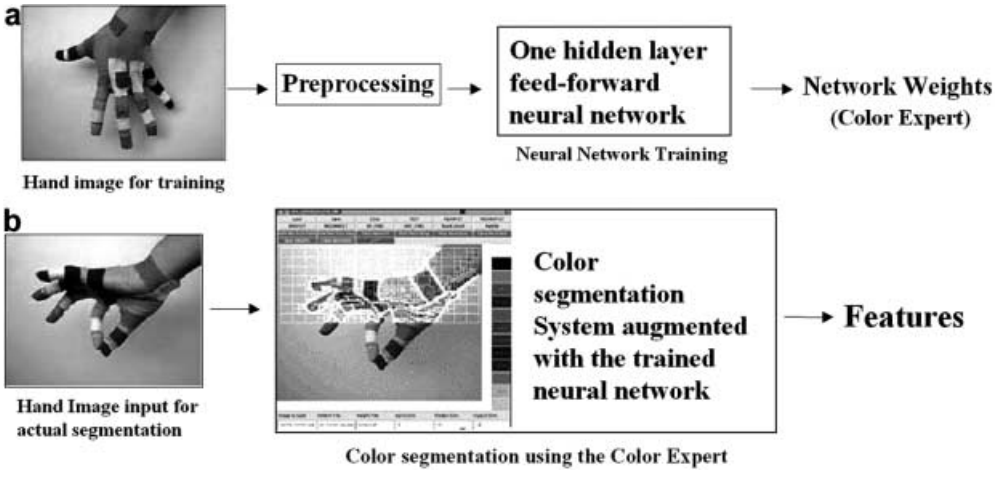

Fig. 9a,b. **a** Training the color expert, based on colored images of a hand whose joints are covered with distinctively colored patches. The trained network will be used in the subsequent phase for segmenting image. **b** A hand image (not from the training sample) is fed to the augmented segmentation program. The color decision during segmentation is done by consulting to the color expert. Note that a smoothing step (not shown) is performed before segmentation

vector with a corresponding confidence vector giving a confidence level for each component in the feature vector. Each finger is marked with two patches of the same color. Sometimes it may not be possible to determine which patch corresponds to the fingertip and which to the knuckle. In those cases the confidence value is set to 0.5. If a color is not found (e.g., the patch may be obscured), a zero value is given for the confidence. If a unique color is found without any ambiguity then the confidence value is set to 1. The segmented centers of regions (color markers) are taken as the approximate articulation point positions. To convert the absolute color centers into a feature vector we simply subtract the wrist position from all the centers found and put the resulting relative (x,y) coordinate into the feature vector (but the wrist is excluded from the feature vector as the positions are specified with respect to the wrist position).

4.3.2 3-D hand model matching. Our model-matching algorithm uses the feature vector generated by the segmentation system to attain a hand configuration and pose that would result in a feature vector as close as possible to the input feature vector (Fig. 10). The scheme we use is a simplified version of Lowe's (1991); see Holden (1997) for a review of other hand recognition studies.

The matching algorithm is based on minimization of the distance between the input feature and model feature vector, where the distance is a function of the two vectors and the confidence vector generated by segmentation system. Distance minimization is realized by hill climbing in feature space. The method can handle occlusions by starting with "don't cares" for any joints whose markers cannot be clearly distinguished in the current view of the hand.

The distance between two feature vectors $\boldsymbol{F}$ and $\boldsymbol{G}$ is computed as follows:

$$D(\boldsymbol{F},\boldsymbol{G}) = \sqrt{(F_i - G_i)^2 C_i^f C_i^g}$$

where subscripting denotes components, and C^f, and C^g denote the confidence vectors associated with $\boldsymbol{F}$ and $\boldsymbol{G}$. Given this result of the visual processing – our *hand-shape recognition* schema – we can clearly read off the following components of the hand state: $a(t)$; aperture of the virtual fingers involved in grasping; and $o_3(t)$, $o_4(t)$, the two angles defining how close the thumb is to the hand as measured relative to the side of the hand and to the inner surface of the palm (see Fig. 4).

The remaining components can easily be computed once the object affordance and location is known. The computation of the components $d(t)$, $v(t)$, $o_1(t)$, and $o_2(t)$, defined in *Sect. 2.2*, constitute the tasks of the *hand–object spatial relation analysis* schema and the *hand-motion detection* schema. These require visual inspection of the relation between hand and target, and visual detection of wrist motion, respectively. It is clear that they pose only minor challenges for visual processing compared with those we have solved in extracting the hand configuration. We thus have completed our exposition of the (nonbiological) implementation of visual analysis of hand state. Section 5.3 presents a justification of the visual analysis of hand state schema by showing MNS1 performance when the hand state was extracted by the described visual recognition system based on a real video sequence. However, when we turn to modeling the core mirror circuit (grand schema 3) in Sect. 4.4 we will not use this implementation of visual analysis of hand state but instead, to simplify computation, we will use synthetic output generated by the reach/grasp simulator to emulate the values that could be extracted with this visual system. Specifically, we use the hand/grasp simulator to produce both (i) the visual appearance of such a movement for our inspection (Fig. 7 left), and (ii) the hand-state trajectory associated with the movement (Fig. 7, right). Especially for training we need to generate and process too many grasp actions, which makes it impractical to use the visual processing system without special hardware as the computational time requirement is too high. Nevertheless, we need to show the similarity of the data from the visual system and the simulator: we have already shown that the grasp simulator generates aperture and velocity profiles that are similar to those in real grasps. Of course, there is still the question of how well the our visual system can extract these features and, more importantly, how similar are the other components of the hand state that we did not specifically craft to match the real data. Preliminary positive evidence will be presented in Sect. 5.3.

4.4 Grand schema 3: core mirror circuit

As illustrated in Fig. 6b, our detailed analysis of the core mirror circuit does not require simulation of visual analysis of hand state and of reach and grasp so long as we ensure that it receives the appropriate inputs. Thus, we supply the object affordance and grasp command

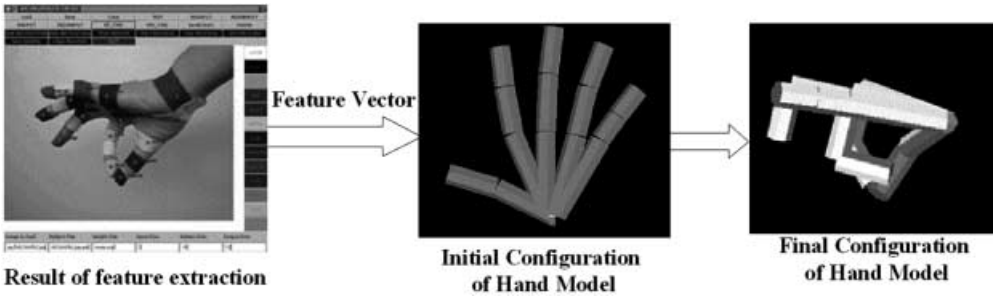

Fig. 10. Illustration of the model-matching system. *Left*: markers located by feature extraction schema. *Middle* and *right*: initial and final stages of model matching. After matching is performed a number of parameters for the hand configuration are extracted from the matched 3-D model

directly to the network at each trial. (Actually, we conduct experiments to compare performance with and without an explicit input which codes object affordance.) The hand-state input is more interesting. Rather than provide visual input to the visual analysis of hand state schema and have it compute the hand-state input to the core mirror circuit, we use our reach and grasp simulator to simulate the performance of the *observed* primate – and from this simulation we extract (as in Fig. 7) both a graphical display of the arm and hand movement that would be seen by the *observing* monkey, as well as the hand-state trajectory that would be generated in his brain. We thus use the time-varying hand-state trajectory generated in this way to provide the input to the model of the core mirror circuit of the observing monkey without having to simultaneously model his visual analysis of hand state. Thus, we have implemented the core mirror circuit in terms of neural networks using as input the synthetic data on hand state that we gather from our reach and grasp simulator (however, see Sect. 5.3 for a simulation with real data extracted by our visual system). (Figure 13 shows an example of the recognition process together with the type of information supplied by the simulator.)

4.4.1 Neural network details. In our implementation, we used a feedforward neural network with one hidden layer. In contrast to the previous sections, we can here identify the parts of the neural network as Fig. 5 schemas in a one-to-one fashion. The hidden layer of the model neural network corresponds to the o*bject affordance–hand state association schema*, while the output layer of the network corresponds to the *action recognition schema* (i.e., we identify the output neurons with the F5 mirror neurons). In the following formulation ***MR*** (mirror response) represents the output of the *action recognition schema* and ***MP*** (motor program) denotes the target of the network [copy of the output of *motor program* (*grasp*) *schema*]. ***X*** denotes the input vector applied to the network, which is the transformed hand state (and the object affordance). The transformation applied is described in Sect. 4.4.2. The learning algorithm used is back propagation (Rumelhart et al. 1986) with a momentum term. The formulation is adapted from Hertz et al. (1991).

Activity propagation (the forward pass) is given by

$$\boldsymbol{MR}_i = g\left(\sum_j \mathbf{W}_{ij} g\left(\sum_k w_{jk}\mathbf{X}_k\right)\right)$$

Learning weights from the input layer to the hidden layer are given by

$$\mathbf{W}_{ij} = \mathbf{W}_{ij} + \eta(t)\delta\mathbf{W}_{ij} + \mu\mathbf{W}_{ij}^{\text{old}},$$

where

$$\delta\mathbf{W}_{ij} = g'\left(\sum_j \mathbf{W}_{ij} g\left(\sum_k w_{jk}\mathbf{X}_k\right)\right)(\boldsymbol{MP}_i - \boldsymbol{MR}_i)$$

$$\mathbf{W}_{\text{old}} = \mathbf{W}$$

Learning weights from the hidden layer to the output layer are given by

$$\mathbf{w}_{jk} = \mathbf{w}_{jk} + \eta(t)\delta\mathbf{w}_{jk} + \mu\mathbf{w}_{jk}^{\text{old}}$$

where

$$\delta\mathbf{w}_{jk} = g'\left(\sum_k \mathbf{w}_{jk}\mathbf{X}_k\right)\mathbf{X}_k$$

$$\mathbf{w}_{jk}^{\text{old}} = \mathbf{w}_{jk}$$

The squashing function g we used was $g(x) = 1/(1+e^{-x})\eta$ and μ are the learning rate and the momentum coefficient, respectively. In our simulations, we adapted η during training such that if the output error was consistently decreasing then we increased η; otherwise we decreased η. We kept μ as a constant set to 0.9. W is the $3 \times (6+1)$ matrix of real numbers representing the hidden-to-output weights. w is the $6 \times (210+1)$ [$6 \times (220+1)$ in the explicit affordance coding case] matrix of real numbers representing the input to hidden weights, and X is the $210+1$ ($220+1$ in the explicit affordance coding case)- component input vector representing the hand state (trajectory) information. (The extra "1" comes from the fact that the formulation we used hides the bias term required for computing the output of a unit in the incoming signals as a fixed input clamped to 1.)

4.4.2 Temporal-to-spatial transformation. The input to the network was formed in a way to allow encoding of temporal information without the use of a dynamic neural network, and solved the scaling problem. The input at any time represented the entire input from the start of the action until the present time t. To form the input vector, each of the seven components of the hand-state trajectory to time t is fitted by a cubic spline (see Kincaid and Cheney 1991 for a formulation), and the splines are then sampled at 30 uniformly spaced intervals. The hand-state input is then a vector with 210 components: 30 samples from the time-scaled spline fitted to the seven components of the hand-state time series. Note then that no matter what fraction t is of the total time T of the entire trajectory, the input to the network at time t comprises 30 samples of the hand state uniformly distributed over the interval [0, t]. Thus the sampling is less densely distributed across the trajectory to date as t increases from 0 to T.

An alternative approach would be to use a simple-recurrent-neural-network style architecture to recognize hand-state trajectories. However, this raises an extra quantization or segmentation step to convert the continuous hand-state trajectories to discrete states. With our approach, we avoid this extra step because the quantization is implicitly handled by the learning process.

For MNS1, we chose to use the spline-based time-to-space transformation, deferring the investigation of models based on recurrent networks (but not necessarily simple recurrent networks) to our later development of

neurally realistic models of the finer-grain schemas of Fig. 5.

Figure 11 demonstrates the preprocessing we use to transform time varying hand-state components into spatial code. In the figure only a single component (the aperture) is shown as an example. The solid curve indicates the available information when the 66% of the grasp action is completed. In reality a digital computer (and thus the simulator) runs in discrete time steps, so we construct the continuous curve by fitting a cubic spline to the collected samples for the value represented (the aperture value in this case). Then we resample 30 points from the (solid) curve to form a vector of size 30. In effect, this presents the network with the stretched spline shown by the dotted curve in Fig. 11. This method has the desirable property of avoiding the time scaling problem to establish the equivalence of actions that last longer than shorter ones, as it is the case for a grasp for an object far from to the hand compared to a grasp to a closer object. By comparing the dotted curve (what the network sees at $t = 0.66$) with the "solid plus dashed" curve (the overall trajectory of the aperture), we can see how much the network's input is distorted. As the action gets closer to its end the discrepancy between the curves tends to zero. Thus, our preprocessing gives rise to an approximation to the final representation when a certain portion or more of the input is seen. Figure 12 samples the temporal evolution of the spatial input the network receives.

4.4.3 Neural network training. The training set was constructed by making the simulator perform various grasps in the following way.

1. The objects used were a cube of changing size (a generic size cube scaled by a random scale factor between 0.5 and 1.5), a disk (approximated as a thin prism), and a ball (approximated as a dodecahedron), again scaled randomly by a number between 0.75 and 1.5. In this particular trial, we did not change the disk size. In the training set formation, a certain object always received a certain grasp (unlike the testing case).
2. The target locations were chosen from the surface patches of a sphere centered on the shoulder joint. The patch is defined by bounding meridian (longitude) and parallel (latitude) lines. The extent of the meridian and parallel lines was from $-45°$ to $45°$. The step chosen was $15°$. Thus the simulator made $7 \times 7 = 49$ grasps per object. The unsuccessful grasp attempts were discarded from the training set.
3. For each successful grasp, two negative examples were added to the training set in the following way. The inputs (a group of 30) for each parameter are randomly shuffled. In this way, the network was forced to learn the order of activity within a group rather than learning the averages of the inputs (note that the shuffling does not change mean and variance). The second negative pattern was used to stress that the distance to target was important. The target location was perturbed and the grasp was repeated (to the original target position).
4. Finally, our last modification in the backpropagation training algorithm was to introduce a random input pattern (totally random; no shuffling) on the fly during training and ask the network to produce zero output for those patterns. In this way we not only biased the network to be as silent as possible during ambiguous input presentation but also gave the network a greater chance of reaching global minima.

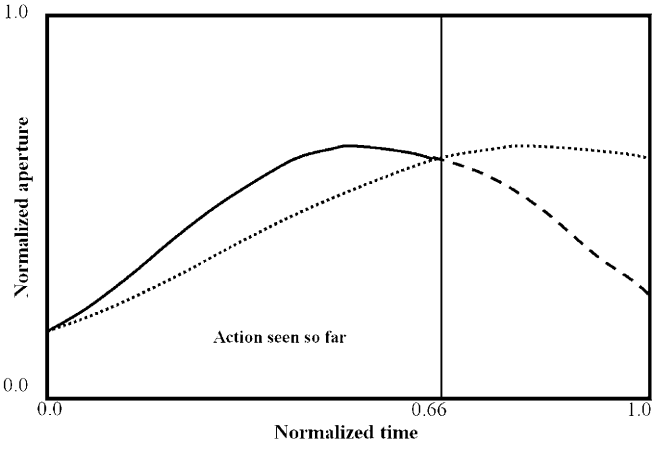

Fig. 11. The scaling of an incomplete input to form the full spatial representation of the hand state As an example, only one component of the hand state—the aperture—is shown. When the 66% of the action is completed, the preprocessing we apply effectively causes the network to receive the stretched hand state (the *dotted curve*) as input as a rerepresentation of the hand state information accessible to that time (represented by the *solid curve*; the *dashed curve* shows the remaining, unobserved part of the hand state)

It should be emphasized that the network was trained using the complete trajectory of the hand state (analogous to adjusting synapses after the self-grasp is completed). During testing, in contrast, the prefixes of a trajectory were used (analogous to the predictive response of mirror neurons while observing a grasp action). The network thus yielded a time-course of activation for the mirror neurons. As we shall see in Sect. 5, initial prefixes yield little or no mirror neuron activity, and ambiguous prefixes may yield transient activity of the "wrong" mirror neurons.

We thus need to make two points to highlight the contribution of this study:

1. It is, of course, trivial to train a network to pair complete trajectories with the final grasp type. What is interesting here is that we can train the system on the basis of final grasp but then observe the whole time course of mirror neuron activity, yielding predictions for neurophysiological experiments by highlighting the importance of the *timing* of mirror neuron activity.
2. Again, it is commonly understood that the training method used here, namely backpropagation, is not intended to be a model of the cellular learning mechanisms employed in cerebral cortex. This might be a matter of concern were we intending to model the

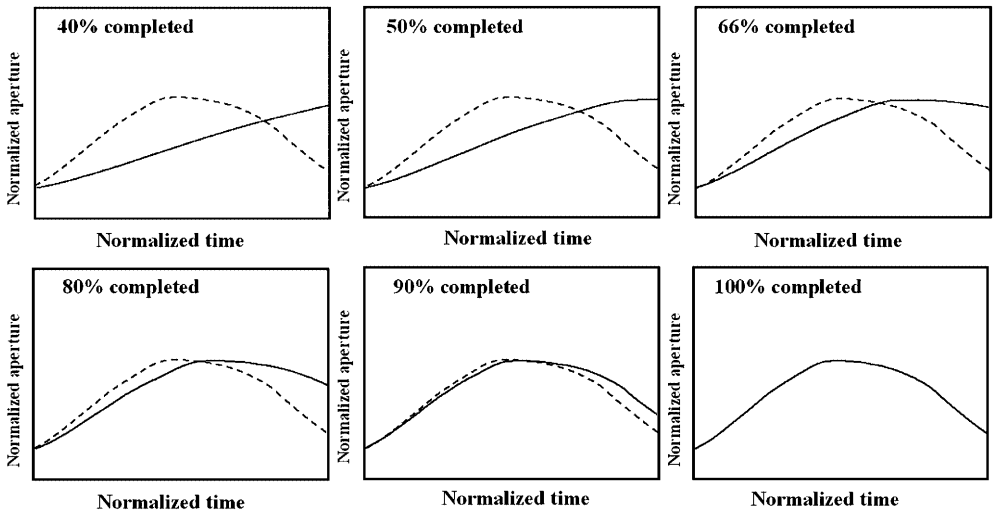

Fig. 12. The *solid curve* shows the effective input that the network receives as the action progresses. At each simulation cycle the scaled curves are sampled (30 samples each) to form the spatial input for the network. Towards the end of the action the networks input gets closer to the final hand state

time course of learning, or analyze the effect of specific patterns of neural activity or neuromodulation on the learning process. However, our aim here is quite different: we want to show that the connectivity of mirror neuron circuitry can be established through training, and that the resultant network can exhibit a range of novel, physiologically interesting behaviors during the process of action recognition. Thus, the actual choice of training procedure is purely a matter of computational convenience, and the fact that the method chosen is nonphysiological does not weaken the importance of our predictions concerning the timing of mirror neuron activity.

5 Results

In this study we experimented with two types of network. The first has only the hand state as the network input. We call this version the *nonexplicit affordance coding network* since the hand state will often imply the object affordance in our simple grasp world. The second network we experimented with – the *explicit affordance coding network* – has affordance coding as one set of its inputs. The number of hidden layer units in each case was chosen to be six, and there were three output units, each one corresponding to a recognized grasp.

5.1 Nonexplicit affordance coding experiments

We first present results with the MNS1 model implemented without an explicit object affordance input to the core mirror circuit. We then study the effects of supplying an explicit object affordance input.

5.1.1 Grasp resolution. In Fig. 13, we let the (trained) model observe a grasp action. Figure 13a demonstrates the executed grasp by giving the views from three different angles to show the reader the 3-D trajectory traversed. Figure 13b shows the extracted hand state (left) and the response of the (trained) core mirror network (right). In this example, the network was able to infer the correct grasp without any ambiguity, since a single curve corresponding to the observed grasp reaches a peak and the other two units' outputs are close to zero during the whole action. The horizontal axis for both figures is such that the onset of the action and the completion of the grasp are scaled to 0 and 1, respectively. The vertical axis in the hand-state plot represents a normalized (min = 0, max = 1) value for the components of the hand state, whereas the output plot represents the average firing rate of the neurons (no firing = 0, maximum firing = 1). The plotting scheme that is used in Fig. 13 will be used in later simulation results as well.

It is often impossible (even for humans) to classify a grasp at a very early phase of the action. For example, the initial phases of a power grasp and precision grasp can be very similar. Figure 14 demonstrates this situation where the model changes its decision during the action and finally reaches the correct result towards the end of the action. To create this result we used the "outer limit" of the precision grasp by having the model perform a precision grasp for a wide object (using the wide opposition axis). Moreover, the network had not been trained using this object for precision grasp. In Fig. 14b, the curves for power and precision grips cross towards the end of the action, which shows the change of decision of the network.

5.1.2 Spatial perturbation. We next analyze how the model performs if the observed grasp action does not meet the object. Since we constructed the training set to stress the importance of distance from hand to object, we expected that network response would decrease with increased perturbation of target location.

Figure 15 shows an example of such a case. However, the network's performance was not homogeneous over the workspace: for some parts of the space the network would yield a strong mirror response even with comparatively large perturbation. This could be

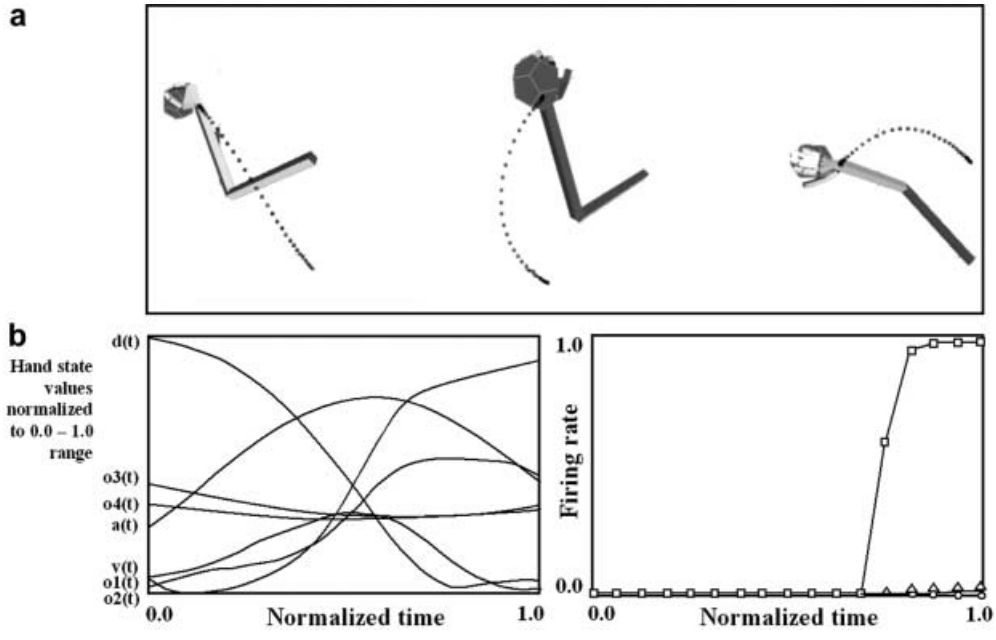

Fig. 13a,b. **a** A single grasp trajectory viewed from three different angles to clearly show its 3-D pattern. The wrist trajectory during the grasp is marked by spatial markers, with the distance between any two consecutive traveled in equal time intervals. **b** *Left*: The input to the network. Each component of the hand state is labelled. *Right*: How the network classifies the action as a power grasp: *squares*, power-grasp output; *triangles*, precision grasp; *circles*, side-grasp output. Note that the response for precision and side grasp is almost zero

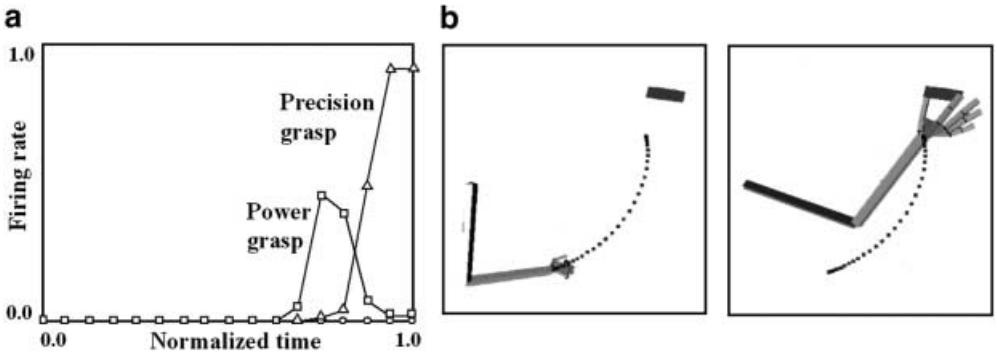

Fig. 14a,b. Power and precision grasp resolution. The conventions used are as in Fig. 13. **a** The curves for power and precision cross towards the end of the action showing the change of decision of the network. **b** The *left* panel shows the initial configuration, and the *right* panel shows the final configuration of the hand

due to the small size of the training set. However, interestingly the network's response had some specificity in terms of the direction of the perturbation. If the object's perturbation direction were similar to the direction of hand motion then the network would be more likely to disregard the perturbation (since the trajectory prefix would then approximate a prefix of a valid trajectory) and signal a good grasp. Note that the network reduces its output rate as the perturbation increases, however the decrease is not linear and after a critical point it sharply drops to zero. The critical perturbation level also depends on the position in space.

5.1.3 Altered kinematics. Normally, the simulator produces bell-shaped velocity profiles along the trajectory of the wrist. In our next experiment, we tested action recognition by the network for an aberrant trajectory generated with constant arm-joint velocities. The change in the kinematics does not change the path generated by the wrist. However the trajectory (i.e., time course along the path) is changed and the network is capable of detecting this change (Fig. 16). The notable point is that the network acquired this property without our explicit intervention (i.e., the training set did not include any negative samples for altered velocity profiles). This is because the input to the network at any time comprises 30 evenly spaced samples of the trajectory up to that time. Thus, changes in velocity can change the pattern of change exhibited across those 30 samples. The extent of this property is again dependent on spatial location.

It must be stressed that all the virtual experiments presented in this section used a single trained network; no new training samples were added to the training set for any virtual experiment.

5.1.4 Grasp and object axes mismatch. The last virtual experiment we present with nonexplicit affordance coding explores the model's behavior when the object opposition axis does not match the hand opposition axis. This example emphasizes that the response of the network is affected by the opposition axis of the object being grasped. Figure 17 shows the axis orientation change for the object and the effect of this perturbation on the output of the network. The arm simulator first performed a precision grasp to a thin cylinder. The response of the mirror neuron model to this action observation is shown in Fig. 17, leftmost panel. As can be seen from the plot, the network confidently activated the mirror neuron coding precision grip. The middle panel shows the output of the network when the object

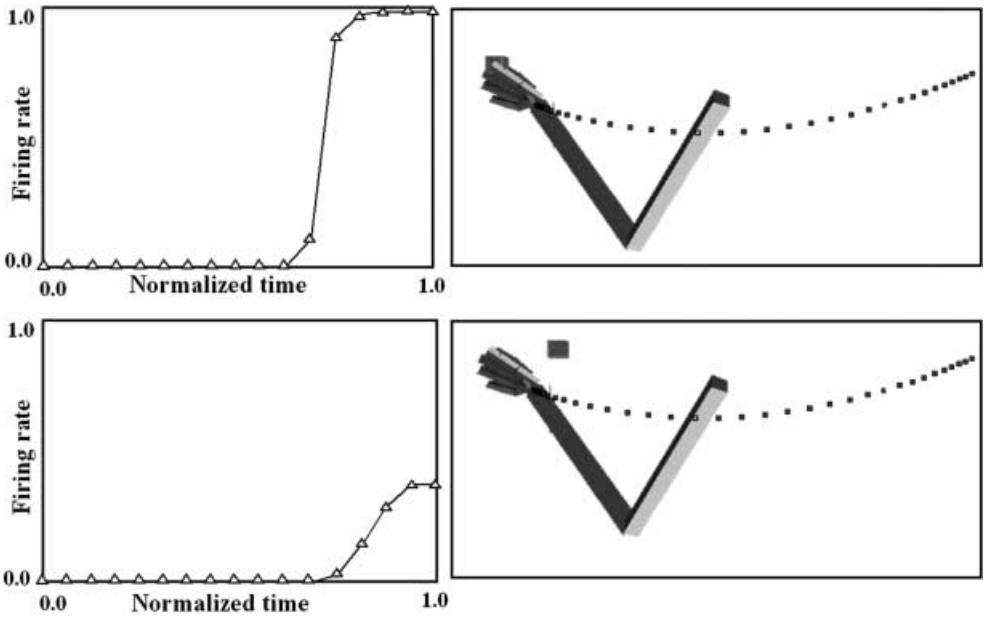

Fig. 15. *Top*: strong precision grip mirror response for a reaching movement with a precision pinch. *Bottom*: Spatial location perturbation experiment. The mirror response is greatly reduced when the grasp is not directed at a target object (Only the precision grasp related activity is plotted; the other two outputs are negligible.)

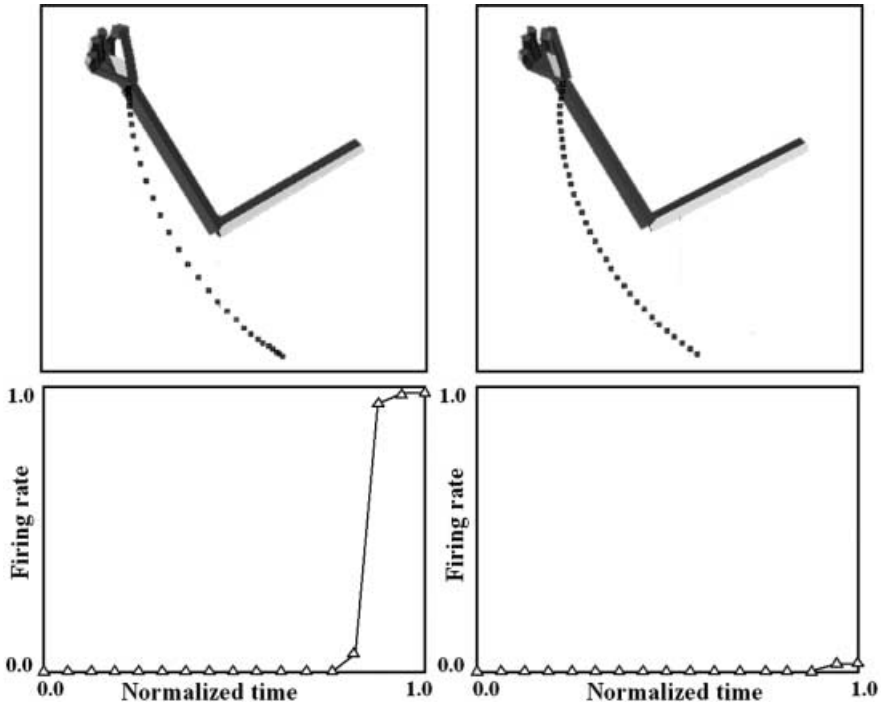

Fig. 16. Altered kinematics experiment. *Left*: the simulator executes the grasp with bell-shaped velocity profile. *Right*: the simulator executes the same grasp with constant velocity. *Top row* shows the graphical representation of the grasps and the *bottom row* shows the corresponding output of the network (Only the precision grasp related activity is plotted; the other two outputs are negligible.)

is changed to a flat plate but the kinematics of the hand is kept the same. The response of the network declined to almost zero in this case. This is an extreme example – the objects in Fig. 17 (rightmost panel) have opposition axes 90° apart, enabling the network to detect the mismatch between the hand (action) and the object. With less change in the new axis the network would give a higher response and, if the opposition axis of the objects were coincident, the network would respond to both actions (with different levels of confidence depending on other parameters).

5.2 Explicit affordance coding experiments

Now we switch our attention to the explicit affordance coding network. Here we want to see the effect of object affordance on the model's behavior. The new model is similar to that given before except that it not only has inputs encoding the current prefix of the hand state trajectory (which includes hand–object relations), but also has a constant input encoding the relevant affordance of the object under current scrutiny. Thus, both the training of the network and the performance of the

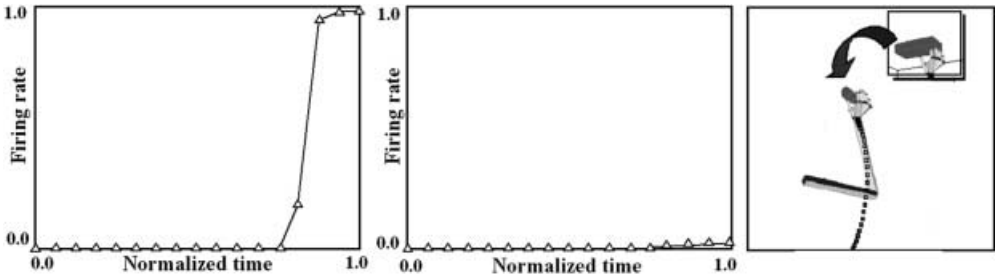

Fig. 17. Grasp and object axes mismatch experiment. *Right*: the change of the object from cylinder to a plate (an object axis change of 90°). *Left*: the output of the network before the change (the network turns on the precision grip mirror neuron). *Middle*: the output of the network after the object change (Only the precision grasp related activity is plotted; the other two outputs are negligible.)

trained network will exhibit effects of this additional, affordance input.

Due to the simple nature of the objects studied here, the affordance coding used in the present study only encodes the object size. In general, one object will have multiple affordances. The ambiguity then would be solved using extra cues such as the contextual state of the network. We chose a coarse coding of object size with ten units. Each unit has a preferred value; the firing of a unit is determined by the difference of the preferred value and the value being encoded. The difference is passed through a non-linear decay function by which the input is limited to the 0 to 1 range (the larger the difference, the smaller the firing rate). Thus, the explicit affordance coding network has 220 inputs (210 hand state inputs, plus 10 units coarse coding the size). The number of hidden layer units was again chosen as six and there were again three output units, each one corresponding to a recognized grasp.

We have seen that the MNS1 model without explicit affordance input displayed a biasing effect of object size in Sect. 5.1.1; the network was biased toward power grasp while observing a wide precision pinch grasp (the network initially responded with a power-grasp activity even though the action was a precision grasp). The model with full affordance replicates the grasp resolution behavior seen in Fig. 12. However, we can now go further and ask how the temporal behavior of the model with explicit affordance coding reflects the fact that object information is available throughout the action. Intuitively, one would expect that the object affordance would speed up the grasp resolution process (which is actually the case, as will be shown in Fig. 19).

In Sects. 5.2.1 and 5.2.2 we look at the effect of affordance information in two cases: (i) where we study the response to precision pinch trajectories appropriate to a range of object sizes, and (ii) where on each trial we use the same time-varying hand-state trajectory but modify the object affordance part of the input. In each case, we are studying the response of a network that has been previously trained on a set of normal hand-state trajectories coupled with the corresponding object affordance (size) encoding.

5.2.1 Temporal effects of explicit affordance coding. To observe the temporal effects of having explicit coding of affordances to the model, we choose a range of object sizes, and then for each size drive the (previously trained) network with both affordance (object size) information and the hand-state trajectory appropriate for a precision pinch grasp appropriate to that size of object. For each case we looked at the model's response. Figure 18 shows the resultant level of mirror responses for four cases (tiny, small, medium, and big objects). The filled squares indicate the precision activity while the empty squares indicate the power-grasp-related activity. When the object to be grasped is small, the model turns on the precision mirror response more quickly and with no ambiguity (Fig. 18, top panels). The vertical bar drawn at time 0.6 shows the temporal effect of object size (affordance). The curves representing the precision grasps are shifted towards the end (time = 1), as the object size gets bigger. Our interpretation is that the model gained the property of predicting that a small object is more likely to be grasped with a precision pinch rather than a power pinch. Thus the larger the object, the more of the trajectory that had to be seen before a confident estimation could be made that it was indeed leading to a precision pinch. In addition, as we indicated earlier, the explicit affordance coding network displays the grasp resolution behavior during the observation of a precision grip being applied to large objects (Fig. 18, bottom panels: right panel and, to a lesser degree, the left panel).

We also compared the general response time of the nonexplicit affordance coding implementation with the explicit coding implementation. The network with affordance input is faster to respond than the previous one. Moreover, it appears that – when affordance and grasp type are well correlated – having access to the object affordance from the beginning of the action not only lets the system make better predictions but also smoothes out the neuron responses. Figure 19 summarizes this: it shows the precision response of both the explicit and nonexplicit affordance case for a tiny object (dashed and solid curves, respectively).

5.2.2 Teasing apart the hand state and object affordance components. We now look at the case where the hand-state trajectory is incompatible with the affordance of the observed object. In Fig. 20, the upper-left panel shows the system output for a precision grasp directed to a medium-sized object whose affordance is supplied to the network. We then repeatedly input the hand-state trajectory generated for this particular action but in each

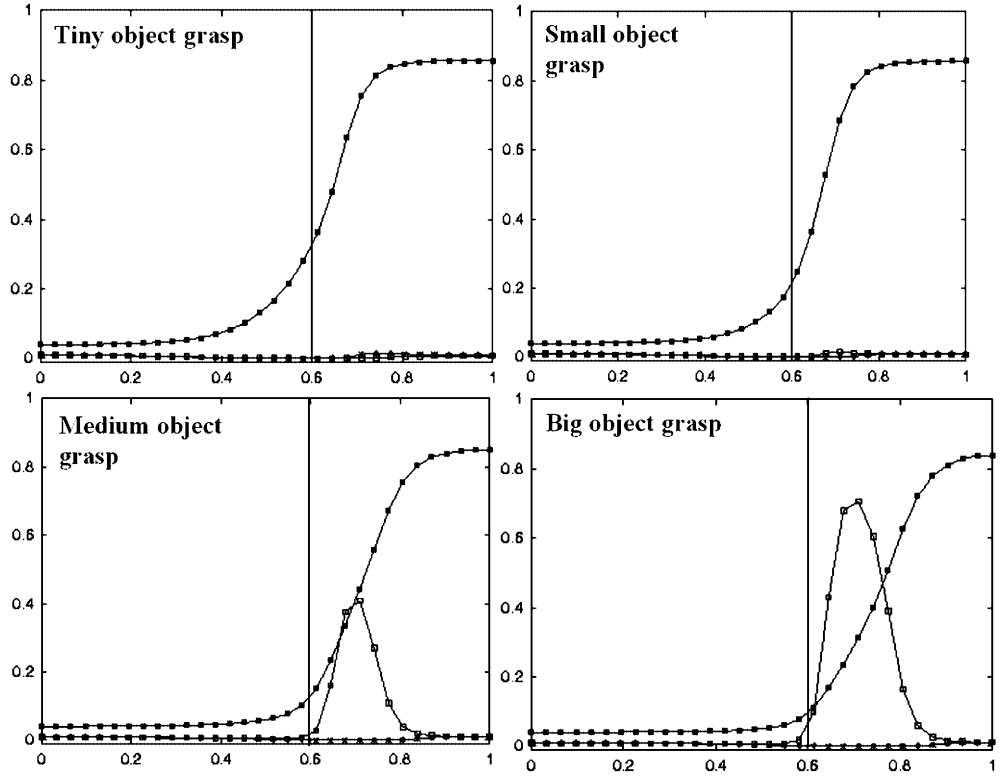

Fig. 18. The plots show the level of mirror responses of the explicit affordance coding object for an observed precision pinch for four cases (tiny, small, medium, and big objects). The *filled squares* indicate the precision activity while the *empty squares* indicate the power-grasp-related activity

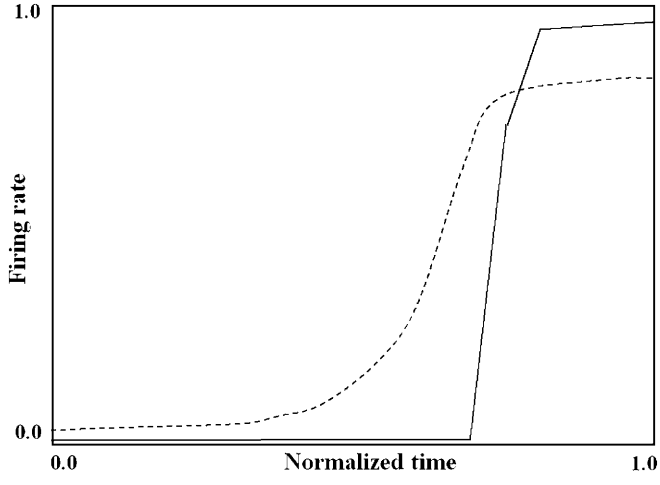

Fig. 19. *Solid curve*: the precision grasp output, for the nonexplicit affordance case, directed to a tiny object. *Dashed curve*: the precision grasp output of the model to the explicit affordance case, for the same object

trial use an object affordance discordant with the observed trajectory affordance (i.e., using a reduced or increased size of the object). The plots in Fig. 20 show the change of the output of the model due to the change in the affordance. The results shown in these plots tell us two things. First, the recognition process becomes fuzzier as the object gets bigger, because the larger object sizes biases the network towards the power grasp. In the extreme case the object affordance can even overwhelm the hand state and switch the network decision to power grasp (Fig. 20, lower right panel). Moreover, for large objects, the large discrepancy between the observed hand state trajectory and the size of the objects results in the network converging on a confident assessment for neither grasp.

Secondly, the resolution point (the crossing point of the precision and power curves) shows an interesting temporal behavior. It may be intuitive to think that as the object gets smaller the network's precision decision gets quicker and quicker (similar to what we have seen in Sect. 5.2.1). However, although this is the case when the object is changing size from big to small, it is not the case when the object size is getting medium to tiny (i.e., the crossing time has a local minimum between the two extreme object sizes, as opposed to being at the tiny-object extreme). Our interpretation is that the network learned an implicit parameter related to the absolute value of the difference of the hand aperture and the object size such that the maximum firing is achieved when the difference is smallest; that is, when the hand trajectory matches best with the object. This will explain why the network has the quickest resolution for a size between the biggest and the smallest sizes.

Figure 21 shows the time of resolution versus object size in graphical form. We emphasize that the model easily executes the grasp recognition task when the hand-state trajectory matches object affordance. We do not include all the results of these control trials, as they are similar to the cases mentioned in Sect. 5.1.

5.3 Justifying the visual analysis of hand state schema

Before closing, we would like to present a simulation run using a real video input to justify our claim that hand

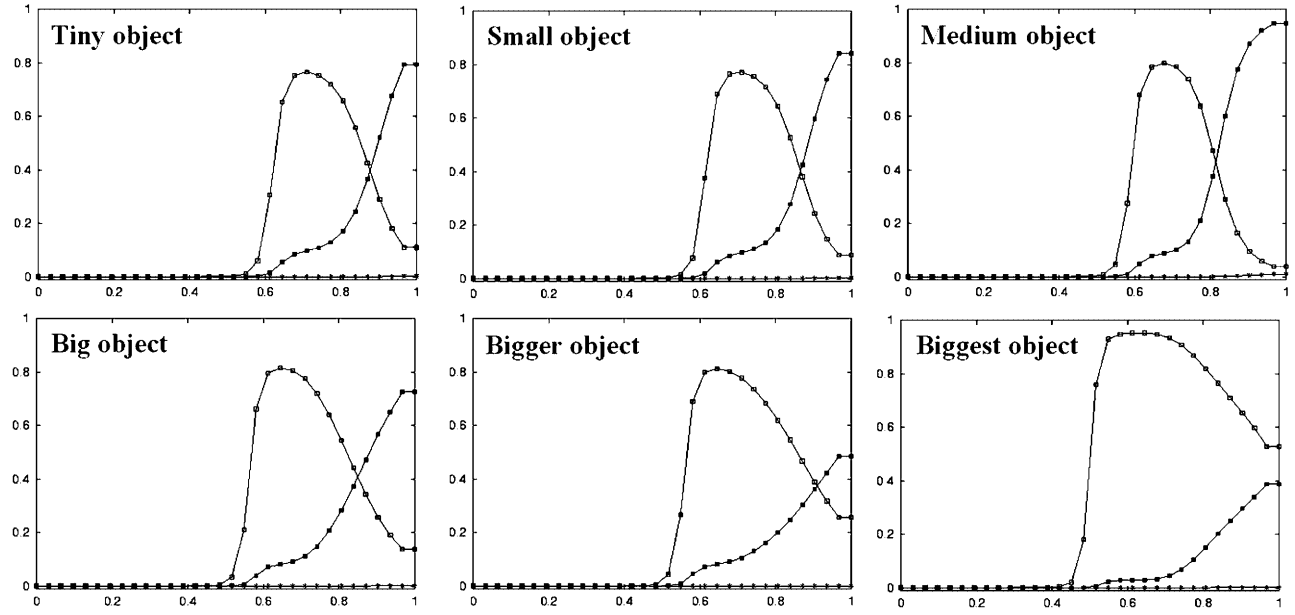

Fig. 20. *Filled squares* indicate the precision-grasp-related cell activity, while the *empty squares* represent the power-grasp-related cell activity. The grasps show the effect of changing the object affordance while keeping a constant hand-state trajectory. In each case, the hand-state trajectory provided to the network is appropriate to the medium-sized object, but the affordance input to the network encodes the size shown. In the case of the biggest object affordance, the effect is enough to overwhelm the hand-state's precision bias

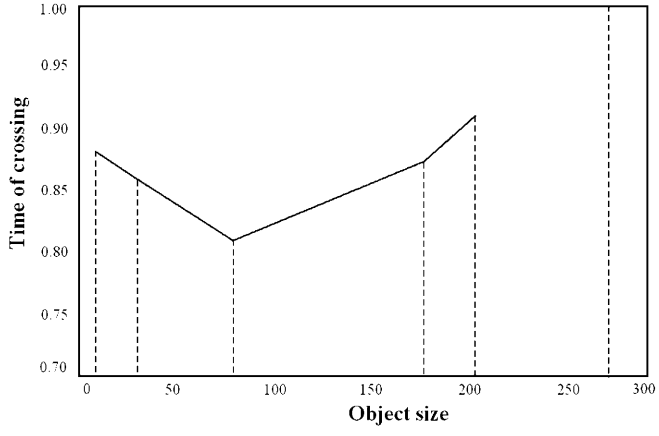

Fig. 21. The graph is drawn to show the decision switch time versus object size. The minimum is not at the boundary, that is, the network will detect a precision pinch quickest with a medium object size. Note that the graph does not include a point for "Biggest object" since there is no resolution point in this case (see the lower right panel of Fig. 20)

Fig. 22. The precision-grasp action used to test our visual system is depicted by superimposed frames (not all the frames are shown)

state can be extracted from real video and used to drive the core mirror circuit. The object affordances are supplied manually since we did not address object recognition in our visual system. However, the rest of the hand state is extracted by the hand recognition system as described in Sect. 4.3. Figure 22 depicts the precision-grasp action used as input video for the simulation.

The result of 3-D hand matching is illustrated in Fig. 23. The color extraction is performed as described in Sect. 4.3.1, but is not shown in the figure. It would be very rewarding to perform all our MNS1 simulations using this system. However, the quality of the video equipment available and the computational power requirements did not allow us to collect many grasp examples to train the core mirror circuit. Nevertheless, we did test the hand state extracted by our visual system from this real video sequence on the MNS1 model that has already been trained with the synthetic grasp examples.

Figure 24 shows the recognition result when the actual visual recognition system provided the hand state based on the real video sequence shown in Fig. 23. Although the output of the network did not reach a high level of confidence for any grasp type, we can clearly see that the network favored the precision grasp over the side and power grasps. It is also interesting to note that a similar competition (this time between side- and precision-grasp outputs) took place as we saw (Fig. 14) when the grasp action was ambiguous.

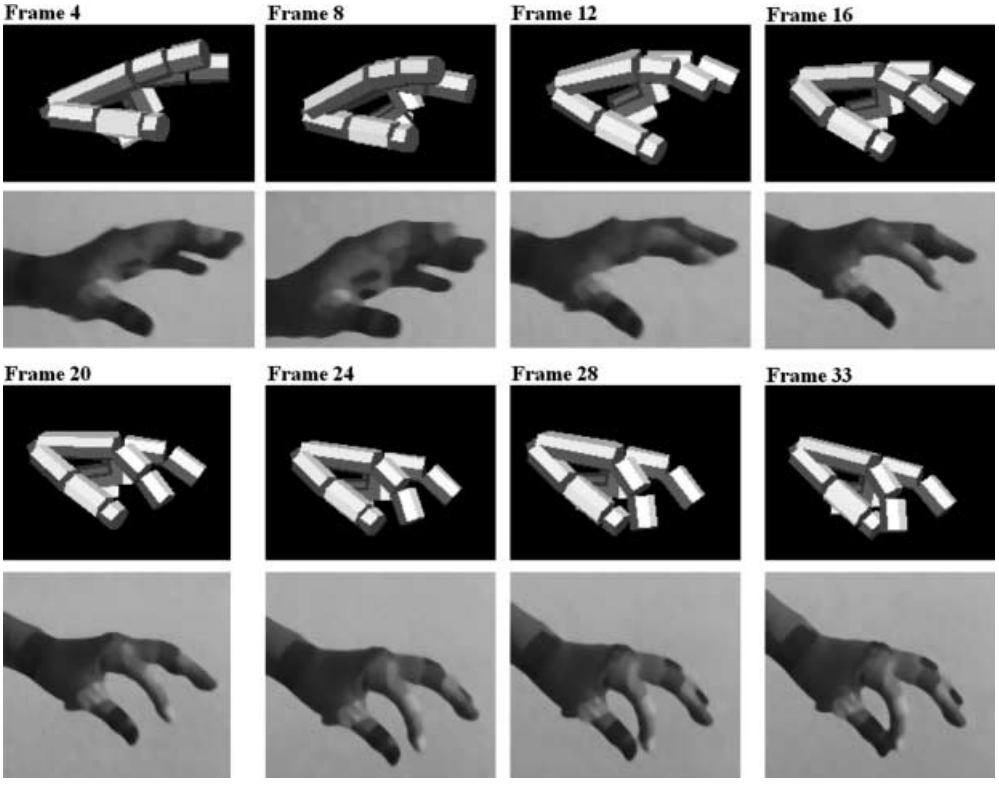

Fig. 23. The video sequence used to test the visual system is shown together with the 3-D hand-matching result (over each frame). Again, not all the frames are shown

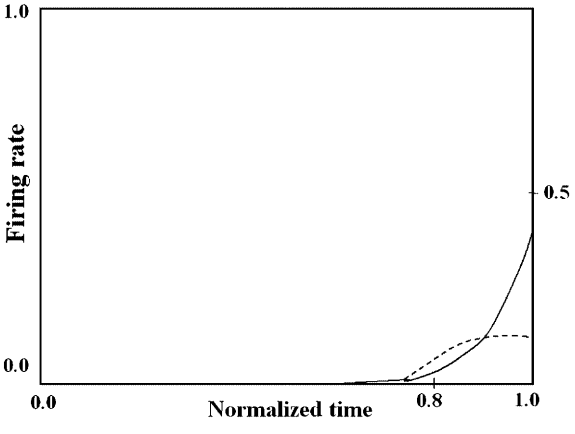

Fig. 24. The output of the MNS1 model when driven by the visual recognition system while observing the action depicted in Fig. 22. It must be emphasized that the training was performed using the synthetic data from the grasp simulator, whereas testing is performed using the hand state extracted by the visual system only. *Dashed line*, side grasp related activity; *solid line*, precision grasp related activity. Power-grasp activity is not visible as it coincides with the time axis

6 Discussion

6.1 The hand-state hypothesis

Because the mirror neurons within monkey premotor area F5 fire not only when the monkey performs a certain class of actions but also when the monkey observes similar actions, it has been argued that these neurons are crucial for understanding of actions by others. We agree with the importance of this role and indeed have built upon it elsewhere, as we now briefly discuss. Rizzolatti et al. (1996b) used a positron-emission tomography study to show that both grasping observation and object prehension yield highly significant activation in the rostral part of Broca's area (a significant part of the human language system) as compared to the control condition of object observation. Moreover, Massimo-Matelli (in Rizzolatti and Arbib 1998) demonstrated a homology between monkey area F5 and area 45 in the human brain (Broca's area comprises areas 44 and 45). Such observations led Rizzolatti and Arbib (1998), building on Rizzolatti et al. (1996a), to formulate *the mirror-system hypothesis:* human Broca's area contains a mirror system for grasping which is homologous to the F5 mirror system of monkey, and this provides the evolutionary basis for language parity; i.e., for an utterance to mean roughly the same for both speaker and hearer. This adds a neural "missing link" to the tradition that roots speech in a prior system for communication based on manual gesture.

Arbib (2001) then refines this hypothesis by showing how evolution might have bridged from an ancestral mirror system to a "language-ready" brain via increasingly sophisticated mechanisms for imitation of manual gestures as the basis for similar skills in vocalization and the emergence of protospeech. In some sense, then, the present paper can be seen as extending these evolutionary concerns back in time. Our central aim was to give a computational account of the monkey mirror system by asking (i) what data must the rest of the brain supply to the mirror system? and (ii) how could the mirror system learn the right associations between classification of its own movements and the movement of others? In seeking

to ground the answer to (i) in earlier work on the control of hand movements (Iberall and Arbib 1990), we were led to extend our evolutionary understanding of the mirror system by offering *the hand-state hypothesis:* the basic functionality of the F5 mirror system is to elaborate the appropriate feedback – what we call the *hand state* – for opposition-space-based control of manual grasping of an object. Given this functionality, the social role of the F5 mirror system in understanding the actions of others may be seen as an exaptation gained by generalizing from one's own hand to another's hand.

The hand-state hypothesis provides a new explanation of the evolution of the "social capability" of mirror neurons, hypothesizing that these neurons first evolved to augment the "canonical" F5 neurons (active during self-movement but not specifically during the observation of grasping by others) by providing visual feedback on "hand state," relating the shape of the hand to the shape of the object.

6.2 Neurophysiological predictions

We introduced the MNS1 model of F5 and related brain regions as an extension of the FARS model of circuitry for visually guided grasping of objects that links parietal area AIP with F5 canonical neurons. The MNS1 model illustrated in Fig. 5 includes hypotheses as to how different brain regions may contribute to the functioning of the mirror system, and the region-by-region analysis of neural circuitry remains a target for current research and future publications. However, the implementation here took a different approach, aggregating these regions into three "grand schemas" – visual analysis of hand state, reach and grasp, and the core mirror circuit – for each of which we present a detailed implementation. To justify the claim that the model exhibits neurophysiologically interesting behaviors, we must look more carefully at the structure of the implementation, stressing that it is only the activity of mirror neurons in the core mirror circuit for which we make this claim. We developed the visual analysis of hand state schema simply to the point of demonstrating algorithms powerful enough to take actual video input of a hand (though we simplified the problem somewhat by using colored patches) and produce hand-state information. The reach and grasp schema then represented all the functionality for taking the location and affordance of an object, and determining the motion of a hand and arm to grasp it. However, the aim *in the present paper* was not to model the neural mechanisms involved in these processes. Instead, we showed that if we used the reach and grasp schema to generate an observed arm–hand trajectory (i.e., to represent the reach and grasp generator of the monkey or human being observed), then that simulation could directly supply the corresponding hand-state trajectory, and we thus use these data so that we can analyze the core mirror circuit schema (Fig. 6b) in isolation from the visual analysis of hand state. However, note that we have also justified the visual analysis of hand state schema by showing in a simulation that the core mirror circuit can be driven with the visual system without any synthetic data from the reach and grasp schema.

Moreover, this hand-state input (regardless of being synthetic or real) was presented to the network in a way which avoids the use of a dynamic neural network. To form the input vector, each of the seven components of the hand-state trajectory, up to the present time t, is fitted by a cubic spline. Then this spline is sampled at 30 uniformly spaced intervals; i.e., no matter what fraction t is of the total time T of the entire trajectory, the input to the network at time t comprises 30 samples of the hand state uniformly distributed over the interval $[0, t]$. The network is trained using the full trajectory of the hand state in a specific grasp; the training set pairs each such hand-state history as input with the final grasp type as output. On the contrary, when testing the model with various grasp observations, the input to the network was the hand-state trajectory that was available up to that instant. This exactly parallels the way the biological system (the monkey) receives visual (object and hand) information: when the monkey performs a grasp, the learning can take place after the observation of the complete (self-) generated visual stimuli. On the other hand, in the observation case the monkey mirror system predicts the grasp action based on the partial visual stimuli (i.e., before the grasp is completed). The network thus yields a time course of activation for the mirror neurons, yielding predictions for neurophysiological experiments by highlighting the importance of the *timing* of mirror neuron activity. We saw that initial prefixes will yield little or no mirror neuron activity, and ambiguous prefixes may yield transient activity of the "wrong" mirror neurons.

Since our aim was to show that the connectivity of mirror neuron circuitry can be established through training, and that the resultant network can exhibit a range of novel, physiologically interesting behaviors during the process of action recognition, the actual choice of training procedure is purely a matter of computational convenience, and the fact that the method chosen, namely back propagation, is nonphysiological does not weaken the importance of our predictions concerning the timing of mirror neuron activity.

With this we turn to neurophysiological predictions made in our treatment of the core mirror circuit, namely the "grounding assumptions" concerning the nature of the input patterns received by the circuit and the actual predictions on the timing of mirror neuron activity yielded by our simulations.

6.2.1 Grounding assumptions. The key to the MNS1 model is the notion of *hand state* as encompassing data required to determine whether the motion and preshape of a moving hand may be extrapolated to culminate in a grasp appropriate to one of the affordances of the observed object. Basically a mirror neuron must fire if the preshaping of the hand conforms to the grasp type with which the neuron is associated; and the extrapolation of *hand state* yields a time at which the hand is grasping the object along an axis for which that affordance is appropriate. What we emphasize here is

not the specific decomposition of the hand state $F(t)$ into the seven specific components [$d(t)$, $v(t)$, $a(t)$, $o_1(t)$, $o_2(t)$, $o_3(t)$, $o_4(t)$] used in our simulation, but rather that the input neural activity will be a distributed neural code which carries information about the movement of the hand toward the object, the separation of the virtual fingertips, and the orientation of different components of the hand relative to the opposition axis in the object. The further claim is that this code will work just as well for measuring how well another monkey's hand is moving to grasp an object as for observing how the monkey's own hand is moving to grasp the object, allowing self-observation by the monkey to train a system that can be used for observing the actions of others and recognizing just what those actions are.

We provided experiments to compare the performance of the core mirror circuit with and without the availability of explicit affordance information (in this case, the size of the object) to strengthen our claim that it is indeed adaptive for the system to have this additional input available, as shown in Fig. 6b. Note that the "grasp-command" input shown in that figure serves here as a training input, and will, of course, plays no role in the recognition of actions performed by others.

Also we have given a justification of the visual analysis of hand state schema by showing in a simulation that the core mirror circuit can be driven with the visual system we implemented without requiring the reach and grasp simulator to provide synthetic data.

6.2.2 Novel predictions. Experimental work to date has tended to emphasize the actions to be correlated with the activity of each individual mirror neuron, while paying little attention to the temporal dynamics of mirror neuron response. By contrast, our simulations make explicit predictions on how a given (hand-state trajectory, affordance) pair will drive the time course of mirror neuron activity – with non-trivial response possibly involving the activity of other mirror neurons in addition to those associated with the actual grasp being observed. For example, a grasp with an ambiguous prefix may drive the mirror neurons in such a way that the system will, in certain circumstances, at first give weight to the wrong classification, with only the late stages of the trajectory sufficing for the incorrect mirror neuron to be vanquished.

To obtain this prediction we created a scene where the observed action consisted of grasping a wide object with a precision pinch (thumb and index finger opposing each other). Usually this grasp is applied to small objects (imagine grasping a pen along its long axis versus grasping it along its thin center axis). The mirror response we got from our core mirror circuit was interesting. First, the system recognized (while the action was taking place) the action as a power grasp (which is characterized by enclosing the hand over large objects; e.g. grasping an apple), but as the action progressed the model unit representing precision pinch started to become active and the power-grasp activity started to decline. Eventually the core mirror circuit settled on the precision pinch. This particular prediction is testable and indeed suggests a whole class of experiments. The monkey has to be presented with unusual or ambiguous grasp actions that require a "grasp resolution." For example, the experimenter can grasp a section of banana using precision pinch from its long axis. Then we would expect to see activity from power-grasp-related mirror cells followed by a decrease of that activity accompanied by increasing activity from precision-pinch-related mirror neurons.

The other simulations we made lead to different testable predictions such as the mirror response in case of a spatial perturbation (showing the monkey a fake grasp where the hand does not really meet the object) and altered kinematics (perform the grasp with different kinematics than usual). The former is in particular a justification of the model, since in the mirror-neuron literature it has been reported that the spatial contact of the hand and the object is almost always required for the mirror response (Gallese et al. 1996). On the other hand, the altered kinematics result predicts that an alteration of the kinematics will cause a decrease in the mirror response. We have also noted how a discrepancy between hand-state trajectory and object affordance may block or delay the system from classifying the observed movement.

In summary, we have conducted a range of simulation experiments – on grasp resolution, spatial perturbation, altered kinematics, temporal effects of explicit affordance coding, and analysis of compatibility of the hand state to object affordance – which demonstrate that the present model is not only of value in providing an implemented high-level view of the logic of the mirror system, but also serves to provide interesting predictions ripe for neurophysiological testing, as well as suggesting new questions to ask when designing experiments on the mirror system. However, we must note that this study has excluded some actions (such as tearing and twisting) for which mirror activity has been observed. As new neurophysiological studies on monkeys expand the range of actions for which the temporal response of the mirror system is delimited, we will expand our model to explain the new findings and suggest yet further classes of experiments to probe the structure and function of the mirror system – as well as increasing the set of brain regions in Fig. 5 for which increasingly realistic neural models are made available.

Acknowledgments. This work was supported in part by a Human Frontier Science Program grant to M.A.A which made possible extensive discussions of data on the mirror system with Giacomo Rizzolatti, Vittorio Gallese, and Massimo Matelli, to whom we express our gratitude. Writing of the paper was also supported in part by an ARC-IREX grant to Robyn Owens for the study of "Hand movement recognition and language: assistive technologies and neural mechanisms" while M.A.A was visiting as adjunct professor in the Computer Science Department at the University of Western Australia.

Appendix: Implementation details

The system was implemented in Java on a Linux operating system. The grasp simulator can be accessed

at http://java.usc.edu/~erhan/sim6.1. The material at this URL also includes the action recognition circuitry. The simulation environment enables the users to test this simplified version of the action recognition ability of the network.

A.1 The segmentation system

The segmentation system as a whole works as follows:

1. Start with N rectangles (called nodes), set thresholds for red, green, and blue variances as rV, gV, bV.
2. For each node calculate the red, green, blue variance as rv, gv, bv.
3. If any of the variance is higher than the threshold (rv > rV or gv > gV or bv > bV), then split the node into four equal pieces and apply steps 2 and 3 recursively.
4. Feed in the mean red, green, and blue values in that region to the color expert to determine the color of the node.
5. Make a list of nodes that are of the same color (add node to the list reserved for that color).
6. Repeat steps 2–5 until no split occurs.
7. Cluster (in terms of Euclidean distance on the image) the nodes and discard the outliers from the list (use the center of the node as the position of the node). The discarding is performed either when a region is very far from the current mean (weighted center) or it is not "connected" to the current center position. The connectedness is defined as follows. The regions A and B are connected if the points lying on the line segment joining the centers of A and B are the same color as A and B. Once again, the color expert is used to determine the percentage of the correct colors (of A and B) lying on the line segment. If this percentage is over a certain threshold (e.g., 70%) then the regions A and B are taken as "connected." (This strategy would not work for a "sausage-shaped" region, but does work for patches created by the coloring we used in the glove.)
8. For each pruned list (corresponding to a color), find the weighted (by the area of the node) mean of the clusters (in terms of image coordinates).
9. Return the cluster mean coordinates as the segmented region's center.

So we do not exactly perform the merge part of the split–merge algorithm. The return values from this procedure are the (x,y) coordinates of the center of color patches found. Another issue is how to choose the thresholds. The variance values are not very critical; a too-small value increases computation time but does not affect the number of colors extracted correctly (though the returned coordinates may be shifted slightly). To see why intuitively, one can notice that the center of a rectangle and the centroid of the centers of the quarter rectangles (say after a split operation) would be the same. This means that if a region is split unnecessarily (because the threshold variances were set to very small values), it is likely to be averaged out by our algorithm since it is likely that the four split rectangles will have the same color and will be connected (with our definition of connectedness).

A.2 Grasp planning and execution for a precision pinch

Precision pinch planning involves the folowing steps:

1. Determine the opposition axis to grasp the object.
2. Compute the two (outer) points A and B at which the opposition axis intersects the object surface. These serve as the contact points for the virtual fingers that will be involved in the grasp.
3. Assign the real fingers to virtual fingers. The particular heuristic we used in the experiments was the following. If the object is on the right (left) with respect to the arm, then the thumb is assigned to the point A if A is on the left of (at a lower level than), B; otherwise the thumb is assigned to B. The index finger is assigned to the remaining point.
4. Determine an approximate target position C, for the wrist. Mark the target for the wrist on the line segment connecting the current position of the wrist and the target for the thumb a fixed length (determined by the thumb length) away from the thumb target.
5. Solve the inverse kinematics for only the wrist reach (ignore the hand).
6. Solve the inverse kinematics for grasping. Using the sum of distance squares of the finger tips to the target contact points, do a random hill-climbing search to minimize the error. Note that the search starts with placing the wrist at point C. However, the wrist position is not included in the error term.
7. The search stops when the simulator finds a configuration with error close to zero (success) or after a fixed number of steps (failure to reach). In the success case the final configuration is returned as the solution for the inverse kinematics for the grasp. Otherwise "failure to reach" is returned.
8. Execute the reach and grasp. At this point the simulator knows the desired target configuration in terms of joint angles. So what remains to be done is to perform the grasp in a realistic way (in terms of kinematics). The simplest way to perform the reach is to linearly change the joint angles from the initial configuration to the target configuration. But this does not produce a bell-shaped velocity profile (nor exactly a constant-speed profile either because of the nonlinearity in going from joint angles to end-effector position). The perfect way to plan an end-effector trajectory requires the computation of the Jacobian. However we are not interested in perfect trajectories as long as the target is reached with a bell-shaped velocity profile. To get the effect it is usually sufficient to modify the idea of linearly changing the joint angles a little bit. We simply modulate the change of time by replacing time with a third-order polynomial that will match our constraints for time (starts at 0 and climbs up to 1 monotonically). Note that we are still working in the joint space and our method may

suffer from the nonlinearity in transforming the joint angles to end-effector coordinates. However, our empirical studies showed that a satisfactory result, for our purposes, could be achieved in this way.

References

Andersen RA, Asanuma C, Cowan WM (1985) Callosal and prefrontal associational projecting cell populations in area 7A of the macaque monkey: a study using retrogradely transported fluorescent dyes. J Comp Neurol 232: 443–455

Andersen RA, Asanuma C, Essick G, Siegel RM (1990) Corticocortical connections of anatomically and physiologically defined subdivisions within the inferior parietal lobule. J Comp Neurol 296: 65–113

Arbib MA (1981) Perceptual structures and distributed motor control. In: Brooks VB (ed) Handbook of physiology – The nervous system II, Motor control, Part 1. American Physiological Society, Bethesda, Md., pp 1449–1480

Arbib MA (2002) The mirror system, imitation, and the evolution of language. In: Nehaniv C, Dautenhahn K (eds) Imitation in animals and artifacts. MIT Press, Cambridge, Mass.

Arbib MA, Érdi P, Szentágothai J (1998) Neural organization: structure, function and dynamics. MIT Press, Cambridge, Mass.

Bota M (2001) Neural homologies: principles, databases and modeling. PhD thesis, Department of Neurobiology, University of Southern California, Los Angeles, Calif.

Boussaoud D, Ungerleider LG, Desimone R (1990) Pathways for motion analysis – cortical connections of the medial superior temporal and fundus of the superior temporal visual areas in the macaque. J Comp Neurol 296: 462–495

Cavada C, Goldman-Rakic PS (1989) Posterior parietal cortex in rhesus monkey: II. Evidence for segregated corticocortical networks linking sensory and limbic areas with the frontal lobe. J Comp Neurol 287: 422–445

Fagg AH, Arbib MA (1998) Modeling parietal–premotor interactions in primate control of grasping. Neural Netw 11: 1277–1303

Gallese V, Fadiga L, Fogassi L, Rizzolatti G (1996) Action recognition in the premotor cortex. Brain 119: 593–609

Gentilucci M, Fogassi L, Luppino G, Matelli M, Camarda R, Rizzolatti G (1988). Functional organization of inferior area 6 in the macaque monkey. I. Somatotopy and the control of proximal movements. Exp Brain Res 71: 475–490

Geyer S, Matelli M, Luppino G, Zilles K (2000) Functional neuroanatomy of the primate isocortical motor system. Anat Embryol 202: 443–474

Gibson JJ (1966) The senses considered as perceptual systems. Houghton Mifflin, Boston

Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. Addison-Wesley, Redwood City, Calif.

Hoff B, Arbib MA (1993) Models of trajectory formation and temporal interaction of reach and grasp. J Mot Behav 25: 175–192

Holden EJ (1997) Visual recognition of hand motion. PhD thesis, Department of Computer Science, University of Western Australia, Perth, Australia

Iberall T, Arbib MA (1990) Schemas for the control of hand movements: an essay on cortical localization. In: Goodale MA (ed) Vision and action: the control of grasping. Ablex, Norwood, NJ, pp 163–180

Jeannerod M, Arbib MA, Rizzolatti G, Sakata H (1995) Grasping objects: the cortical mechanisms of visuomotor transformations. Trends Neurosci 18: 314–320

Jordan MI, Rumelhart DE (1992) Forward models: supervised learning with distal teacher. Cogn Sci 16: 307–354

Karniel A, Inbar GF (1997) A model for learning human reaching movements. Biol Cybern 77: 173–183

Kawato M, Gomi H (1992) A computational model of four regions of the cerebellum based on feedback-error learning. Biol Cybern 68: 95–103

Kawato M, Furukawa K, Suzuki R (1987) A hierarchical neural-network model for control and learning of voluntary movement. Biol Cybern 57: 169–185

Kincaid D, Cheney W (1991) Numerical analysis: Mathematics of Scientific Computing Brooks/Cole, Pacific Grove, Calif.

Klein Breteler MDK, Gielen SC, Meulenbroek RG (2001) Endpoint constraints in aiming movements: effects of approach angle and speed. Biol Cybern 85: 65–75

Lambert P, Carron T (1999) Symbolic fusion of luminance-hue-chroma features for region segmentation. Pattern Recogn 32: 1857–1872

Lewis JW, Van Essen (2000) Corticocortical connections of visual, sensorimotor, and multimodal processing areas in the parietal loe of the macaque monkey. J Comp Neurol 428: 112–137

Lowe DG (1991) Fitting parameterized three-dimensional models to images. IEEE Trans Pattern Anal Mach Intell 13: 441–450

Luppino G, Murata A, Govoni P, Matelli M (1999) Largely segregated parietofrontal connections linking rostral intraparietal cortex (areas AIP and VIP) and the ventral premotor cortex (areas F5 and F4). Exp Brain Res 128: 181–187

Maioli MG, Squatrito S, Samolsky-Dekel BG, Sanseverino ER (1998) Corticocortical connections between frontal periarcuate regions and visual areas of the superior temporal sulcus and the adjoining inferior parietal lobule in the macaque monkey. Brain Res 789: 118–125

Marteniuk RG, MacKenzie CL (1990) Invariance and variability in human prehension: implications for theory development. In: Goodale MA (ed) Vision and action: the control of grasping. Ablex, Norwood, NJ, pp 163–180

Matelli M, Camarda R, Glickstein M, Rizzolatti G (1986) Afferent and efferent projections of the inferior area 6 in the macaque monkey. J Comp Neurol 251: 281–298

Matelli M, Luppino G, Murata A, Sakata H (1994) Independent anatomical circuits for reaching and grasping linking the inferior parietal sulcus and inferior area 6 in macaque monkey. Soc Neurosci Abstr 20: 404.4

Maunsell JH (1995) The brains visual world: representation of visual targets in cerebral cortex. Science 270: 764–769

Maunsell JH, Van Essen DC (1983) The connections of the middle temporal visual area (MT) and their relationship to a cortical hierarchy in the macaque monkey. J Neurosci 3: 2563–2586

Meulenbroek RG, Rosenbaum DA, Jansen C, Vaughan J, Vogt S (2001) Multijoint grasping movements. Simulated and observed effects of object location, object size, and initial aperture. Exp Brain Res 138: 219–234

Murata A, Fadiga L, Fogassi L, Gallese V, Raos V, Rizzolatti G (1997) Object representation in the ventral premotor cortex (area F5) of the monkey. J Neurophysiol 78: 2226–2230

Ratcliff G (1991) Brain and space: some deductions from clinical evidence. In: Paillard J (ed) Brain and space, Oxford University Press, Oxford, pp 237–250

Rizzolatti G, Arbib MA (1998) Language within our grasp. Trends Neurosci 21: 188–194

Rizzolatti G, Fadiga L (1998) Grasping objects and grasping action meanings: the dual role of monkey rostroventral premotor cortex (area F5). In: Sensory guidance of movement, Novartis Foundation Symposium 218. Wiley, Chichester, pp 81–103

Rizzolatti G, Camarda R, Fogassi L, Gentilucci M, Luppino G, Matelli M (1988) Functional organization of inferior area 6 in the macaque monkey. II. Area F5 and the control of distal movements. Exp Brain Res 71: 491–507

Rizzolatti G, Fadiga L, Gallese V, Fogassi L (1996a) Premotor cortex and the recognition of motor actions. Cogn Brain Res 3: 131–141

Rizzolatti G, Fadiga L, Matelli M, Bettinardi V, Perani D, Fazio F (1996b) Localization of grasp representations in humans by

PET: 1. Observation versus execution. Exp Brain Res 111: 246–252
Rizzolatti G, Luppino G, Matelli M (1998) The organization of the cortical motor system: new concepts. Electroencephalogr Clin Neurophysiol 106: 283–296
Rosenbaum DA, Meulenbroek RG, Vaughan J, Jansen C (1999) Coordination of reaching and grasping by capitalizing on obstacle avoidance and other constraints. Exp Brain Res 128: 92–100
Roy AC, Paulignan Y, Farne A, Jouffrais C, Boussaoud D (2000) Hand kinematics during reaching and grasping in the macaque monkey. Behav Brain Res 117: 75–82
Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing, vol 1: Foundations. MIT Press, Cambridge, Mass., pp 151-193
Russ JC (1998) The image processing handbook. CRC, Boca Raton, Flo.
Sakata H, Taira M, Murata A, Mine S (1995) Neural mechanisms of visual guidance of hand action in the parietal cortex of the monkey. Cereb Cortex 5: 429–438
Sakata H, Taira M, Kusunoki M, Murata A, Tanaka Y (1997) The TINS lecture – The parietal association cortex in depth perception and visual control of action. Trends Neurosci 20: 350–357
Sonka M, Hlavac V, Boyle R (1993) Image processing, analysis and machine vision. Chapman and Hall, London, NY
Stein JF (1991) Space and the parietal association areas. In: Paillard J (ed.) Brain and space. Oxford University Press, Oxford, pp 185–222
Taira M, Mine S, Georgopoulos AP, Murata A, Sakata H (1990) Parietal cortex neurons of the monkey related to the visual guidance of hand movement. Exp Brain Res 83: 29–36
Wolpert DM, Ghahramani Z (2000) Computational principles of movement neuroscience. Nature Neurosci 3: 1212–1217